

**ANÁLISIS, MANTENIMIENTO, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DE NUEVAS FUNCIONALIDADES Y REINGENIERÍAS RELACIONADAS CON
NUEVOS TIPOS DE ACTIVIDADES EVALUABLES, CARTELERA, CÁLCULO
DE NOTAS Y CARGA Y DESCARGA DE ARCHIVOS, EN EL AULA VIRTUAL
DEL PORTAL WEB COMUNIDAD ACADÉMICA**

CLAUDIA XIMENA DIAZ GARCIA

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2017

**ANÁLISIS, MANTENIMIENTO, DISEÑO, DESARROLLO E IMPLEMENTACIÓN
DE NUEVAS FUNCIONALIDADES Y REINGENIERÍAS RELACIONADAS CON
NUEVOS TIPOS DE ACTIVIDADES EVALUABLES, CARTELERA, CÁLCULO
DE NOTAS Y CARGA Y DESCARGA DE ARCHIVOS, EN EL AULA VIRTUAL
DEL PORTAL WEB COMUNIDAD ACADÉMICA**

CLAUDIA XIMENA DIAZ GARCIA

**Trabajo de grado para optar al título de
Ingeniera de Sistemas**

Director

Msc. LUIS IGNACIO GONZÁLEZ RAMÍREZ

Magíster en Informática

**UNIVERSIDAD INDUSTRIAL DE SANTANDER
FACULTAD DE INGENIERÍAS FISICOMECAÑICAS
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA
BUCARAMANGA**

2017

AGRADECIMIENTOS

A DIOS por darme salud, fortaleza y una gran familia.

A mis padres que siempre me han apoyado en mis estudios desde muy pequeña y que a pesar de las dificultades se esforzaron por darme lo mejor.

A mis hermanos que han sido un ejemplo a seguir para mí.

Al profesor Luis Ignacio González por la paciencia y la disposición de atender mis preguntas y dificultades.

A mis compañeros del grupo CALUMET que me ayudaron a solucionar problemas y con los que compartí buenos momentos.

A mi novio Duvan por acompañarme siempre, llenarme de apoyo y sacarme una sonrisa en cualquier momento.

A mis compañeros de la selección de Karate y a mi entrenador Juan José que me enseñaron a no darme por vencida ante las dificultades, tanto en la vida como en el deporte, a luchar por las metas y saber ganar y perder, esas fueron las cosas que me han permitido salir adelante en todos los ámbitos de mi vida.

CONTENIDO

INTRODUCCIÓN	13
1. PLANTEAMIENTO DEL PROYECTO	14
1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL	14
1.2 JUSTIFICACIÓN	14
1.3 OBJETIVOS	15
1.4 IMPACTO Y VIABILIDAD	16
2. MARCO TEÓRICO	18
2.1 ARQUITECTURA CLIENTE/SERVIDOR	18
2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS	22
2.3 BASES DE DATOS	24
2.4 NETBEANS	27
2.5 SISTEMA DE CONTROL DE VERSIONES	28
2.6 PROGRAMACIÓN UTILIZADA	29
2.7 SERVIDORES WEB	31
3. MARCO METODOLÓGICO	32
3.1 PROTOTIPO EVOLUTIVO	32
3.2 LENGUAJE DE MODELADO UNIFICADO	34
3.3 ESTÁNDARES DE PROGRAMACIÓN	37

4.	DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO	40
4.1	PROTOTIPO ESPERADO	40
4.2	DIAGRAMAS DE CASOS DE USO	44
4.3	DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA	49
4.4	DISEÑO Y ANÁLISIS	56
4.5	DESCRIPCIÓN DE LAS ENTIDADES.....	57
4.6	MODELO DE PROCESOS DEL SISTEMA	59
4.7	IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES.....	59
4.8	MANTENIMIENTO Y ADMINISTRACIÓN	60
5.	PRUEBAS DEL SISTEMA	62
5.1	PRUEBAS DE VERIFICACIÓN.....	62
6.	CONCLUSIONES	68
7.	RECOMENDACIONES	69
	BIBLIOGRAFÍA	70
	ANEXOS	72

LISTA DE TABLAS

Tabla 1 - Casos de uso: Crear servicio para nuevos tipos de planes	49
Tabla 2 - Casos de uso: Creación de nuevos tipos de planes.	49
Tabla 3 - Casos de uso: Asignar actividades a nuevos tipos de planes	50
Tabla 4 - Casos de uso: Estudiante visualiza planes.....	50
Tabla 5 - Casos de uso: Administración y creación de nuevos tipos de planes.....	51
Tabla 6 - Casos de uso: Interacción docente – estudiante	51
Tabla 7 - Casos de uso: Creación y visualización de las notas en cartelera	52
Tabla 8 - Casos de uso: Compartir notas entre usuarios.....	52
Tabla 9 - Casos de uso: Creación y visualización de los archivos compartidos	53
Tabla 10 - Casos de uso: Compartir archivos entre usuarios	54
Tabla 11 - Casos de uso: Modificar porcentajes de actividades	54
Tabla 12 - Casos de uso: Asignar planes al crear aula.....	55
Tabla 13 - Descripción de las entidades	58
Tabla 14 - Pruebas Realizadas: Creación de servicio nuevos tipos de planes.....	63
Tabla 15 - Pruebas Realizadas: Creación de nuevos tipos de planes.	63
Tabla 16 - Pruebas Realizadas: Asignar actividades a nuevos tipos de planes	64
Tabla 17 - Pruebas Realizadas: Estudiante visualiza planes.....	64
Tabla 18 - Pruebas Realizadas: Creación y visualización de las notas en cartelera	65
Tabla 19 - Pruebas Realizadas: Compartir notas entre usuarios.....	65
Tabla 20 - Pruebas Realizadas: Creación y visualización de los archivos compartidos	65
Tabla 21 - Pruebas Realizadas: Modificar porcentajes de actividades	66
Tabla 22 - Pruebas Realizadas: Agregar plan de quices al crear aula	67

LISTA DE FIGURAS

Figura 1 - Modelo Cliente/Servidor	18
Figura 2 - Modelo de acceso a JSP	24
Figura 3 - Prototipo evolutivo	32
Figura 4 - Diagramas de Casos de Uso	35
Figura 5 - Diagrama de Secuencias.....	37
Figura 6 - Diagrama de casos de uso: Crear servicio para nuevos tipos de planes.	45
Figura 7- Diagrama de casos de uso: Creación de nuevos tipos de planes.	45
Figura 8 - Diagrama de casos de uso: Asignar actividades a nuevos tipos de planes.	45
Figura 9 - Diagrama de casos de uso: Estudiante visualiza planes.	46
Figura 10 - Diagrama de casos de uso: Administración y creación de nuevos tipos de planes.	46
Figura 11 - Diagrama de casos de uso: Interacción docente – estudiante.	46
Figura 12 - Diagrama de casos de uso: Creación y visualización de las notas en cartelera.....	47
Figura 13 - Diagrama de casos de uso: Compartir notas entre usuarios.	47
Figura 14 - Diagrama de casos de uso: Creación y visualización de los archivos compartidos.	47
Figura 15 - Diagrama de casos de uso: Compartir archivos entre usuarios.	48
Figura 16 - Diagrama de casos de uso: Modificar porcentajes de actividades.	48
Figura 17 - Diagrama de casos de uso: Asignar planes al crear aula.....	48
Figura 18 - Diagrama E/R: Creación de nuevos tipos de planes	56
Figura 19 - Diagrama E/R: Notas en el Aula virtual	56
Figura 20 - Diagrama E/R: Compartir archivos en el Aula virtual.....	56
Figura 21 - Diagrama E/R: Asignar porcentajes a las actividades	57
Figura 22. Diagrama de secuencia: Creación de nuevos tipos de planes	72
Figura 23. Diagrama de secuencia: Asignar actividades a nuevos tipos de planes	73
Figura 24. Diagrama de secuencia: Eliminación de tipo de plan.....	74
Figura 25. Diagrama de secuencia: Creación de las notas en cartelera	75
Figura 26. Diagrama de secuencia: Eliminar notas.....	76
Figura 27. Diagrama de secuencia: Creación de archivos compartidos	77
Figura 28. Diagrama de secuencia: Eliminar archivos compartidos.....	78
Figura 29. Diagrama de secuencia: Asignar porcentajes de actividades	79
Figura 30. Diagrama de secuencia: Crear aula	80

LISTA DE ANEXOS

Anexo A. Modelo de Procesos del Sistema	72
Anexos.....	Ver anexos en la carpeta adjunta al CD

RESUMEN

TÍTULO: Análisis, mantenimiento, diseño, desarrollo e implementación de nuevas funcionalidades y reingenierías relacionadas con nuevos tipos de actividades evaluables, cartelera, cálculo de notas y carga y descarga de archivos, en el Aula Virtual del Portal Web 'Comunidad Académica'.*

AUTOR: Claudia Ximena Díaz García **

PALABRAS CLAVE: Calumet, Portal Web, notas, noticias, planes, archivos, cormoran.

DESCRIPCIÓN:

En la actualidad las escuelas cuentan con un sistema de información orientado a la Web que se encarga de la administración y control de las diferentes actividades tanto académicas, como administrativas que se realizan dentro de las escuelas y las Facultades, así como del control de usuarios y servicios que se les proporcionan.

Actualmente el portal cuenta con el servicio de Aula virtual. Este es un medio en donde el docente complementa sus clases e interactúa con los estudiantes, algunos de los métodos usados para esta comunicación son la cartelera virtual y los archivos compartidos, sin embargo, estos servicios solo permiten publicarse a todo el grupo. Se hace necesario cambiar la interfaz de estos servicios por una actual, cambiar la cartelera virtual por notas en donde no solo el profesor puede publicar, también estudiantes, y ampliar la forma de compartir no sólo a todo el grupo, además a subgrupos y personas.

El aula virtual soporta seis tipos de actividades, entre esas se encuentran talleres, exámenes, trabajos, laboratorios, foros y exposiciones. Se hace necesario crear otros tipos de actividades acordes a lo que el docente necesite.

También se permite agregar el plan de quices al crear el aula ya que esta opción no existía.

Finalmente se agrega la opción de modificar los porcentajes de cada actividad dentro de la nota total de actividades, ya que estas tenían el mismo valor en el total de la nota, esto permite en el caso de existir varios exámenes, por ejemplo, cada uno pueda tener diferente valor, lo que permite personalizar más la forma de calificar del docente.

* Trabajo de grado

** Facultad de Ingenierías Físico-mecánicas. Escuela de Ingeniería de Sistemas e Informática. Director: Msc. Luis Ignacio González Ramírez

ABSTRACT

TITLE: Analysis, maintenance, design, development and implementation of new functionalities and reengineering related to new types of evaluable activities, billboard, calculation of notes and upload and download of files, in the Virtual Classroom of the Web Portal 'Comunidad Académica'. *

AUTHOR: Claudia Ximena Díaz García **

KEY WORDS: Calumet, Web portal, notes, news, plans, archives, cormoran.

DESCRIPTION:

At present the schools have a web oriented information system that is in charge of the administration and control of the different academic and administrative activities carried out within the schools and the Faculties, as well as the control of users and Services provided to them.

Currently the portal has the virtual classroom service. This is a medium in which the teacher complements his classes and interacts with the students, some of the methods used for this communication are the virtual billboard and the shared files, however, these services only allow the whole group to be published. It is necessary to change the interface of these services by a current one, change the virtual billboard by notes where not only the teacher can publish, also students, and expand the way to share not only the whole group, but also subgroups and people.

The virtual classroom supports six types of activities, among which are workshops, exams, jobs, laboratories, forums and exhibitions. It is necessary to create other types of activities according to what the teacher needs.

It is also allowed to add the quizzes plan when creating the classroom since this option did not exist.

Finally, the option of modifying the percentages of each activity within the total score of activities is added, since they had the same value in the total of the note, this allows in the case of several exams, for example, each one can Have different value, which allows to customize more the way of qualifying the teacher.

* Degree Work

** Faculty of Physico-Mechanical Engineering. Department of Systems Engineering and Computing Science. Supervisor Msc. Luis Ignacio González Ramírez

INTRODUCCIÓN

Los Portales Web Comunidad Académica son los principales canales de comunicación e integración entre los miembros de la comunidad de las escuelas y miembros en general de la comunidad UIS. Actualmente los servicios se han extendido a más escuelas de las distintas facultades gracias a la aceptación de los usuarios por lo que se adopta el objetivo de mantener y mejorar los servicios que se ofrecen, además de crear nuevos servicios que satisfagan las necesidades crecientes de los usuarios de la comunidad académica. Por esto es necesario actualizar servicios antiguos y se propone la reingeniería de algunos de estos ofrecidos en los portales web.

Una de las mejoras realizadas esta en soportar la creación de nuevos tipos de planes, debido a que actualmente solo existen seis tipos, estos son: entre esas se encuentran talleres, exámenes, trabajos, laboratorios, foros y exposiciones. Se puede agregar cualquier otro tipo y se creó la interfaz para permitir esto, además se puede modificar su estado entre Activo/Inactivo.

Otra de las mejoras realizadas fue la reingeniería en el Aula a las noticias, notas y archivos, se realizó mejora de la interfaz gráfica, así como de su funcionalidad, brindando más opciones en la forma de compartirse y visualizarse. Además, se realizó reingeniería para que sea posible incluir el plan de quices al crear el aula, algo que no existía antes y que se hace necesario para complementar el manejo de más tipos de actividades. También, se desarrolló la reingeniería que permite asignar porcentajes a las actividades, creando la interfaz necesaria para esto y permitiendo al docente personalizar la forma de calificar sus actividades según lo necesite.

En este documento se presenta y da soporte metodológico, teórico y técnico de las mejoras en los servicios mencionados anteriormente, y del mantenimiento y soporte a usuarios en los portales web.

1. PLANTEAMIENTO DEL PROYECTO

1.1 DEFINICIÓN DE LA SITUACIÓN ACTUAL

Los servicios de los portales Web de las escuelas deben mejorar constantemente y adaptarse a los cambios que se presenten en su entorno, a su vez deben dar solución a los problemas y necesidades que surjan por parte de los usuarios del sistema para incrementar su tiempo de vida útil y no llegar a convertirse en un software obsoleto, razón por la cual las labores de mantenimiento y actualización se hacen indispensables.

El aula virtual y los servicios que esta ofrece deben ser actualizados para soportar nuevas necesidades de la comunidad, la actualización consta de cambio de interfaz e implementación de nuevas funcionalidades que permitan a los usuarios interactuar con los servicios Web, y entre ellos mismos, de una mejor manera. Además, agregar servicios nuevos que permitan al docente tener un manejo más personalizado de la materia y que no lo limite en los tipos de actividades ni en los porcentajes de sus notas.

1.2 JUSTIFICACIÓN

El portal Web Comunidad Académica, en línea con su objetivo de aportar lugares para la comunidad, permite a los docentes y estudiantes encontrarse en el aula virtual y acceder a todos los servicios que esta brinda, algunos de estos servicios son: envío de correos, creación de actividades para que los estudiantes complementen sus conocimientos en la materia, asignación de notas, cartelera virtual, libro del aula, preparación de actividades por semana, entre otros. Se hace necesario cambiar la interfaz de la cartelera virtual y archivos, ya que son servicios en donde el tanto estudiantes como docente pueden compartir información, pero esta es limitada porque solo se permite compartirla a todo el grupo. Por esto es

importante ampliar la forma de compartir la información no sólo al grupo sino además a usuarios individuales y subgrupos de clase, y permitir modificar, borrar, etc. De forma amigable, fácil y eficaz para que los usuarios antiguos se adapten rápido y usuarios nuevos comprendan su funcionamiento fácilmente. Adicional a esto, los archivos y notas se separan dependiendo si son compartidos por el usuario o si son para el usuario, por eso la interfaz precisa de dos pestañas en donde se diferencian estas características para mejorar la interacción con el usuario.

Actualmente el aula cuenta con seis tipos de actividades, estas son: exámenes, talleres, trabajos, foros, exposiciones y laboratorios. No todos los docentes manejan estos tipos de actividades, algunos manejan otros según sus necesidades, pero no había la posibilidad de crear nuevos tipos de actividades así que limitaba el manejo del aula por parte del docente. Por eso se vio la necesidad de crear el servicio para agregar nuevos tipos de actividades, además de permitir activar o desactivar actividades según el uso que el docente le dé, así cada aula podrá tener la cantidad de actividades que necesite sin necesidad de limitar a las seis que había.

Otra necesidad del aula era la de permitir modificar el porcentaje de una actividad dentro de un plan, en el caso de los exámenes, si al semestre se realizaran tres exámenes cada uno tendría el mismo valor dentro de la nota total de la materia. Pero esto se tuvo que modificar ya que pueden existir actividades que tengan más valor que otras según su complejidad, tamaño, tiempo a invertir, etc. Por eso se agregó una interfaz que permitiera modificar el valor de las actividades dentro de la nota de la materia, en el caso de no querer modificar dicho valor entonces seguirá funcionando normalmente, cada actividad con el mismo valor de la nota.

1.3 OBJETIVOS

1.3.1 Objetivo general. Realizar las funciones de mantenimiento, análisis, diseño, desarrollo e implementación de los servicios de notas, cartelera, tipos de planes y archivos en el Aula del Portal Web, con el fin de hacerlos óptimos, eficientes y

sostenibles, permitiendo acceder a la información de manera más sencilla, ágil, óptima y eficaz.

1.3.2 Objetivos específicos

- Creación de nuevas interfaces gráficas y actualización de las interfaces antiguas.
- Soportar la creación de nuevos tipos de planes, debido a que actualmente solo existen cuatro tipos, estos son: foros, trabajos, exposiciones y laboratorios. Debe poderse agregar cualquier otro tipo, y este debe comportarse correcto al activarse y evaluarse.
- Diseñar y desarrollar el servicio que permita modificar el contenido de los planes, como el cambio de estado entre Activo/Inactivo.
- Realizar reingeniería en el Aula a las notas y archivos, que contemple mejora de la interfaz gráfica, así como de su funcionalidad.
- Realizar reingeniería para que sea posible incluir el plan de quices al crear el aula.
- Realizar reingeniería para que sea posible asignar porcentajes a las actividades.

1.4 IMPACTO Y VIABILIDAD

1.4.1 Impacto. Los portales web de las escuelas han sido una herramienta útil para el manejo de la información, por lo tanto, es necesario realizar labores de administración y mantenimiento, para ofrecer al usuario un sitio más confiable.

Se pretende que los procesos que se realizan en las escuelas cada día sean más ágiles, dinámicos, seguros y eficientes, permitiendo una mejor organización de la información, razón por la cual se crean nuevos servicios que se ajusten a las necesidades que puedan surgir.

1.4.2 Viabilidad. La administración del sitio, soporte a usuarios, mantenimiento y desarrollo de nuevos servicios es viable pues se usará software de libre distribución, recurso humano preparado para tal fin, servidores que marchan de manera legal en las escuelas, equipos disponibles y todo el soporte tecnológico necesario para el desarrollo del mismo.

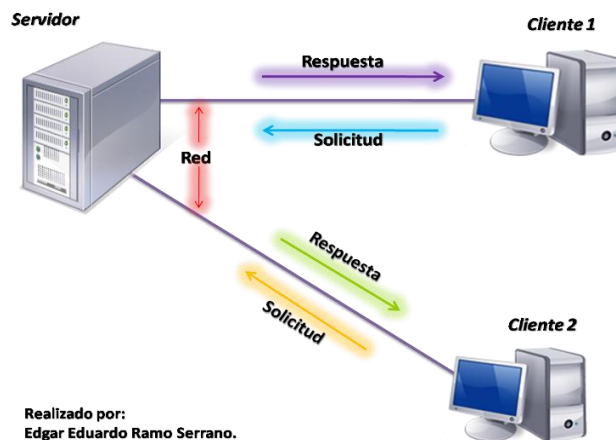
Además, se cuenta con la supervisión por parte del director del proyecto y la colaboración del equipo de trabajo CALUMET, agentes de gran apoyo en la realización de este proyecto.

2. MARCO TEÓRICO

2.1 ARQUITECTURA CLIENTE/SERVIDOR

En la arquitectura cliente/servidor cada uno de los clientes produce un mensaje solicitando un determinado servicio a un servidor (hace una petición) estos envían uno o varios mensajes como respuesta (responden peticiones o provee un servicio). La mayoría del trabajo pesado (procesos de base de datos, procesar la lógica de la aplicación entre otros) está a cargo de los servidores, el cliente comúnmente se encarga de las funciones de administración de la interfaz de usuario, interacción con el usuario, recibir resultados del servidor, generar requerimientos de base de datos, entre otros.

Figura 1 - Modelo Cliente/Servidor



*Fuente: RAMO SERRANO, Edgar Eduardo. Modelo Cliente/ Servidor. [En línea].
(Recuperado 10 de Agosto de 2017) Disponible en
<http://cramercontracramer.blogspot.com.co/2013/12/modelo-cliente-servidor.html>*

2.1.1 Características de la arquitectura Cliente/Servidor. Las características básicas de una arquitectura Cliente / Servidor son:

- Es quien inicia solicitudes o peticiones, tienen por tanto un papel activo en la comunicación.
- El proceso del cliente da la interface entre usuarios y el resto del sistema, maneja recursos compartidos tales como bases de datos, impresoras, módems, etc.
- El cliente y el servidor pueden actuar como una sola entidad y también pueden actuar como entidades separadas, realizando actividades independientes.
- Las tareas del cliente y el servidor tienen diferentes requerimientos como: velocidad del procesador, memoria o capacidad del disco, por tanto, la plataforma de hardware y el sistema operativo del cliente y del servidor no son siempre la misma y eso se conoce como ambiente heterogéneo.
- La escalabilidad horizontal permite agregar más estaciones de trabajo activas sin afectar el rendimiento y la escalabilidad vertical permite mejorar las características del servidor o agregar múltiples servidores.

2.1.2 Clasificación de las arquitecturas Cliente/Servidor. Los sistemas Cliente/Servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece. Se distinguen tres componentes básicos de software para la clasificación:

- **Presentación:** Presentación de resultados al usuario de forma comprensible.
- **Lógica de aplicación:** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Está compuesta por los archivos que contienen los datos persistentes de la aplicación.

Los sistemas Cliente/Servidor se clasifican de acuerdo al nivel de abstracción del servicio que se ofrece. Se distinguen tres componentes básicos de software para la clasificación:

- **Presentación:** Presentación de resultados al usuario de forma comprensible.

- **Lógica de aplicación:** Esta capa es la responsable del procesamiento de la información que tiene lugar en la aplicación.
- **Base de datos:** Está compuesta por los archivos que contienen los datos persistentes de la aplicación.

La siguiente es la clasificación de la arquitectura Cliente/Servidor:

2.1.2.1 Arquitectura Cliente/Servidor de dos capas. Consiste en una capa de presentación y lógica de la aplicación; y otra de la base de datos, cuando el cliente solicita recursos entonces el servidor responde directamente a la solicitud con sus propios recursos. Normalmente esta arquitectura es utilizada en las siguientes situaciones:

- Cuando se requiere poco procesamiento de datos en la organización.
- Cuando se tiene una base de datos centralizada en un solo servidor.
- Cuando la base de datos es relativamente estática.
- Cuando se requiere un mantenimiento mínimo.

2.1.2.2 Arquitectura Cliente/Servidor de tres capas. Define como organizar el modelo de diseño en capas, que pueden estar físicamente distribuidas, es decir que los componentes de una capa solo pueden hacer referencia a componentes en capas inferiores. Este patrón es importante porque simplifica la comprensión y la organización del desarrollo de sistemas complejos, reduciendo las dependencias de forma que las capas más bajas no conscientes de ningún detalle o interfaz de las superiores, está compuesta de:

- Un equipo cliente con una interfaz de usuario (habitualmente se utiliza un navegador web), que solicita los recursos.
- El servidor de aplicaciones (o software intermedio), cuya tarea es prestar los recursos solicitados, pero que requiere de otro servidor para hacerlo.
- El servidor de datos que almacena y proporciona al servidor de aplicaciones los datos que requiere.

2.1.3 Arquitectura Cliente/Servidor aplicada. En el desarrollo de este proyecto se recurre a arquitectura de tres capas, debido a las ventajas ofrecidas como: Escalabilidad, fácil mantenimiento y el manejo de un mayor número de usuarios que la ofrecida por la arquitectura C/S de dos capas. La arquitectura es aplicada de la siguiente forma:

- Capa de Cliente: Interfaz con el usuario, se usa un navegador web.
- Capa Intermedia: Para los servicios del negocio se utiliza un computador configurado como servidor web, el cual almacena el portal web conformado por páginas JSP y JavaBeans. Allí se realizan los procesos complejos, y se solicitan los servicios del servidor de datos cuando es necesario acceder a la información almacenada en la base de datos.
- Capa de Servidor: Se utiliza el motor de bases de datos MySQL, el cual se encuentra en el mismo servidor web.

2.1.4 Ventajas del esquema Cliente/Servidor

- La existencia de plataformas de software y hardware de varios fabricantes y cada vez más a económicas contribuye a la reducción de costos y favorece la flexibilidad en la implantación y actualización de soluciones.
- Este esquema facilita la integración entre sistemas heterogéneos y comparte información permitiendo que las máquinas existentes puedan ser utilizadas con interfaces amigables al usuario, de esta forma integrar los computadores con sistemas medianos y grandes, sin necesidad de que todos tengan que utilizar el mismo sistema operacional.
- Facilita a los diferentes departamentos de una organización soluciones locales, permitiendo la integración de la información principal totalmente.

2.1.5 Desventajas del esquema Cliente/Servidor

- El mantenimiento de los sistemas es complejo pues implica la interacción de diferentes partes hardware y software de diferentes proveedores, lo cual dificulta el diagnóstico de fallas.

- Se cuenta con escasas herramientas para la administración y ajuste del desempeño de los sistemas, además se deben tener estrategias para el manejo de errores y para salvaguardar la consistencia de los datos.
- La seguridad del esquema C/S es preocupante, un ejemplo: las validaciones y verificaciones que se deben hacer tanto en el cliente como en el servidor.
- El desempeño es un aspecto a tener en cuenta en el esquema C/S, problemas de este estilo pueden presentarse por congestión en la red.

2.2 TECNOLOGÍAS DE DESARROLLO DE PÁGINAS WEB DINÁMICAS

Las páginas dinámicas aportan grandes beneficios porque permiten entrar a bases de datos para extraer información que pueda presentarse al usuario, dependiendo de algunos permisos y de la misma forma para almacenar información.

Existen diferentes tecnologías para el desarrollo de páginas dinámicas entre ellas están:

2.2.1 Código del Lado del Cliente (Client Side Scripts). Código ejecutado por los navegadores, el cual los computadores clientes tienen instalados. Las tecnologías más comunes de este tipo son:

- JavaScript: Lenguaje de programación interpretado, es decir, que no requiere compilación, utilizado principalmente en páginas web, con una sintaxis semejante a la del lenguaje Java y el lenguaje C. Permite, crear ventanas, mostrar texto en movimiento y verificar las entradas a un formulario.
- Controles Activos: tecnología Microsoft que permite la creación de aplicaciones Windows, como pueden ser Visual Basic Script o Visual C. Es la respuesta de Microsoft a los Applets de Java.
- Java Applets: Programas escritos en lenguaje de programación Java, se incrustan en HTML y se ejecutan en el navegador gracias a la Máquina Virtual de Java (JVM) que lleva éste incorporado.

2.2.2 Código del Lado del Servidor (Server Side Scripts). Código que se ejecuta en el servidor. Para su actividad el programa ejecuta y procesa los datos o peticiones que el usuario envía desde su navegador, para luego enviar los resultados del programa en una página HTML que el usuario verá normalmente en su navegador. Los más usados son:

- ASP (Active Server Pages): Permite crear dinámicamente páginas Web mediante HTML, scripts, y componentes de servidor ActiveX reutilizables, requiere de un computador configurado como Servidor Web de Microsoft (Microsoft Web Server), el navegador del cliente es indiferente pues el trabajo se realiza del lado del servidor. Da gran uso en la gestión de Bases de Datos ya que puede conectarse a SQL, Access, Oracle u otras.
- PHP (PHP Hypertext Pre-processor): Lenguaje de programación interpretado, diseñado para la creación de páginas web dinámicas. Es un lenguaje de código abierto (Open Source) y gratuito. Su gran potencia se encuentra en la interacción con los motores de bases de datos como Oracle y MySQL.
- JSP (Java Server Pages): tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Permiten la utilización de código Java mediante scripts.

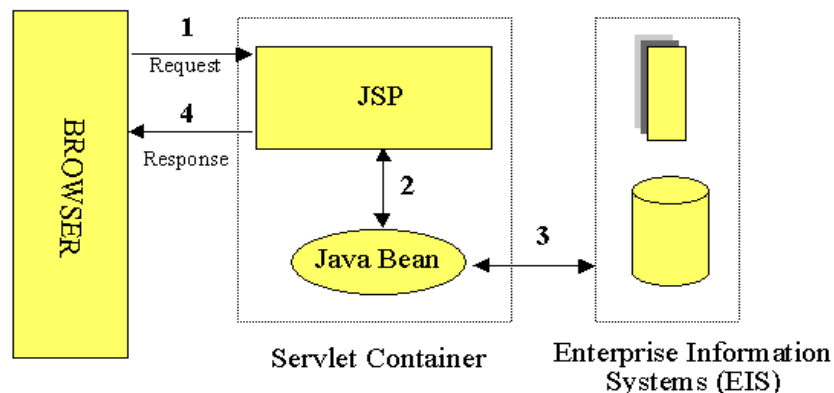
2.2.3 Tecnología aplicada. La tecnología aplicada para la creación del portal web fue JSP, por lo tanto, los nuevos servicios son desarrollados con esta misma tecnología, ya que permite producir aplicaciones independientes de la plataforma y portables a otros sistemas operativos y servidores web.

Una página JSP se compila a una aplicación Java la primera vez que se invoca, y de esta aplicación Java se crea una clase que empieza a ejecutarse en el servidor como un servlet. Un JSP es una página web con etiquetas especiales y código Java incrustado, mientras que un servlet es un programa que recibe peticiones y genera a partir de ellas una página web.

2.2.3.1 Modelo de acceso a JSP

- Un usuario en su navegador web cliente hace una petición que es enviada a un archivo JSP. Este archivo accede a componentes del servidor que generan contenido dinámico y lo presentan en el navegador.
- Después de recibir la petición del cliente, el archivo JSP pide información de un JavaBean si es necesario.
- El JavaBean en turnos puede pedir información de otro JavaBean o de una base de datos.
- Una vez el JavaBean genera el contenido, el archivo JSP puede consultar y presentar el contenido del JavaBean al navegador.

Figura 2 - Modelo de acceso a JSP



La primera vez que un archivo JSP es invocado, este es compilado en un objeto, la respuesta del objeto es HTML estándar, el cual es interpretado por el navegador para ser presentado al usuario. Después de la compilación, el objeto de la página es almacenado en la memoria del servidor. En las peticiones posteriores a esta página, el servidor revisa si el archivo JSP ha cambiado. Si no ha cambiado, el servidor utiliza el objeto de la página compilada guardado en memoria para generar la respuesta al cliente, en caso contrario el servidor automáticamente compila el archivo de la página y reemplaza el objeto en la memoria.

2.3 BASES DE DATOS

Una base de datos es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso, con una redundancia controlada y una estructura que refleja las interrelaciones y restricciones existentes en el mundo real. En la base de datos se almacena información considerada necesaria para una determinada organización o negocio.

Existen modelos que describen la estructura de una base de datos (entidades, atributos y relaciones), la mayoría de los modelos de datos poseen un conjunto de operaciones básicas como consultar y actualizar y eliminar.

2.3.1 Modelos de Bases de Datos. Las bases de datos se clasifican de acuerdo con su modelo de administración de datos. Algunos modelos utilizados con frecuencia son:

2.3.1.1 Base de Datos Jerárquica. Las bases de datos jerárquicas son especialmente útiles en el caso de aplicaciones que manejan un gran volumen de información y datos muy compartidos permitiendo crear estructuras estables y de gran rendimiento. Esta limitado por su incapacidad de representar eficientemente la redundancia de datos.

2.3.1.2 Base de Datos de Red. En este modelo se permite que un mismo nodo tenga varios padres. Ofrece una solución eficiente al problema de redundancia de datos; sin embargo, la dificultad para administrar los datos en una base de datos de red ha conllevado a que sea un modelo usado más por programadores que por usuarios finales.

2.3.1.3 Base de Datos Relacional. Es el más utilizado para modelar problemas reales y administrar datos dinámicamente. Su fundamento es el uso de "relaciones". Estas relaciones podrían considerarse en forma lógica como conjuntos de datos, también llamados tuplas. Cada relación es una tabla que está compuesta por registros (las filas de una tabla), que representan las tuplas, y campos (las

columnas de una tabla). Los datos pueden ser recuperados o almacenados mediante "consultas" que ofrecen una amplia flexibilidad y poder para administrar la información.

2.3.2 Manejadores o Gestores de Bases de Datos. El sistema manejador de bases de datos es la porción más importante del software de un sistema de base de datos. Un DBMS es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de alguna tarea específica.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de Datos.
- Establecer y mantener las trayectorias de acceso a la base de datos de tal forma que los datos puedan ser capturados rápidamente.
- Manejar los datos de acuerdo a las peticiones de los usuarios.
- Registrar el uso de las bases de datos.
- Interacción con el manejador de archivos a través de las sentencias en Lenguaje Manipulador de Datos (Data Manipulation Language, DML) al comando del sistema de archivos.
- Respaldo y recuperación: Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de base de datos.
- Control de concurrencia: consiste en controlar la interacción entre los usuarios concurrentes para preservar la consistencia de los datos.
- Seguridad e Integridad: consiste en contar con mecanismos que permitan el control de la consistencia de los datos evitando que estos se vean perjudicados por cambios no autorizados o previstos.

2.3.3 MySQL. Es un sistema de base de datos operacional considerado uno de los más importantes, utilizado por usuarios del medio para el diseño y programación de

base de datos de tipo relacional. Además MySQL es conocida por desarrollar alta velocidad de búsqueda de datos e información, a diferencia de sistemas anteriores.

2.3.4 Ventajas de MySQL

- El MySQL es un Open Source, o sea código abierto que puede ser usado y modificado.
- Velocidad al realizar las operaciones, lo que le hace uno de los gestores con mejor rendimiento.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consume puede ser ejecutado en una maquina con escasos recursos sin ningún problema.
- Baja probabilidad de corromper datos, incluso si los errores no se producen en el propio gestor, sino en el sistema en el que está.
- Su conectividad, velocidad, y seguridad hacen de MySQL altamente apropiado para acceder a bases de datos en internet.

2.4 NETBEANS

Es un entorno de desarrollo integrado (IDE), siendo una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas escritos en JAVA, pero puede servir para cualquier otro tipo lenguaje de programación. Netbeans es un producto libre y gratuito sin restricciones de uso.

- El Netbeans es un entorno de desarrollo integrado de código abierto escrito completamente en Java usando la plataforma Netbeans, soporta desarrollo de todos los tipos de aplicación Java (J2SE, web, EJB y aplicaciones móviles).
- La versión actual es NetBeans IDE 8.1 Desde NetBeans IDE 6.5 se extienden las características existentes del Java EE (incluyendo Soporte a Persistencia, EEJB 3 y JAX-WS). Adicionalmente, el Netbeans Enterprise Pack soporta el desarrollo de Aplicaciones empresariales java EE 5, incluyendo herramientas de desarrollo

visuales de SOA, herramientas de esquemas XML, orientación a web servicios (for BPEL), y modelado UML.

- Todas las funciones del IDE son provistas por módulos. Cada Módulo provee una función bien definida, tales como el soporte de Java, edición, o soporte para el sistema de control de versiones. Netbeans contiene todos los módulos necesarios para el desarrollo de aplicaciones Java en una sola descarga, permite al usuario comenzar a trabajar inmediatamente.

2.5 SISTEMA DE CONTROL DE VERSIONES

Un sistema de control de versiones es un software que administra el acceso a un conjunto de ficheros, y mantiene un historial de cambios realizados. El control de versiones es útil para guardar cualquier documento que cambie con frecuencia, o el código fuente de un programa.

Normalmente consiste en una copia maestra en un repositorio central, y un programa cliente con el que cada usuario sincroniza su copia local. Además, el repositorio guarda registro de los cambios realizados por cada usuario, y permite volver a un estado anterior en caso de necesidad.

2.5.1 Subversión. Esto permite recuperar versiones antiguas de los datos o examinar cómo han ido evolucionando esto. Su objetivo es ser un sucesor prácticamente compatible del ampliamente usado Concurrent Version system (CVS).

Subversión puede trabajar a través de redes, lo que permite que las personas que estén en diferentes computadores puedan usarlo, con la posibilidad de que varias personas modifiquen y gestionen el mismo conjunto de datos desde sus sitios promueve la colaboración, y como el trabajo está versionado, ya que si se produce algún cambio incorrecto de los datos, sólo hace falta deshacerlo.

2.6 PROGRAMACIÓN UTILIZADA

Para el desarrollo de este proyecto se usó la Programación Orientada a Objetos (P.O.O.). La P.O.O. es una de las formas más populares de programas que usa objetos y sus interacciones para diseñar aplicaciones y programas de computador, intenta simular el mundo real a través del significado de objetos que contienen características y funciones; abstrae algunas características de sistemas naturales complejos como son:

- Atributos: estado del objeto.
- Métodos: comportamiento del objeto.
- Herencia: comportamientos comunes entre objetos relacionados para hallar relaciones de especialización y generalización de comportamientos.

2.6.1 Clases. Definición de todos los elementos de que este hecho un objeto. Cuando se programa un objeto y se definen sus características y funcionalidades, realmente se programa una clase. Por lo tanto, para realizar la abstracción de sistemas naturales, observamos y analizamos un grupo de cosas con características comunes, el resultado de esta abstracción será válido para todas estas cosas.

2.6.2 Objetos. Cualquier cosa real o abstracta, que posee atributos y un conjunto de operaciones que manipulan esos atributos que da un comportamiento particular. Un objeto es una instancia de una clase, el estado del objeto se determina por el estado (valor) de sus propiedades o características (atributos).

2.6.3 Atributos. Características de un objeto siendo un conjunto de datos (valores) y calificadores para aquellos datos. Estos atributos pueden ser desde tipos de datos simples (enteros, caracteres, cadenas de texto) hasta otros objetos.

2.6.4 Métodos. Son funciones o procedimientos propios de la clase que pueden tener acceso a los atributos de la misma para realizar las operaciones para los que son programados.

2.6.5 Herencia. Se fundamenta en usar una clase ya creada para tomar sus características en clases más especializadas o derivadas de ésta para reutilizar el código que sea común con la clase base, y solamente definir nuevos métodos o redefinir algunos de los existentes para ajustarse al comportamiento particular de esta subclase.

2.6.6 Beneficios de la Programación Orientada a Objetos

- Permite obtener aplicaciones modificables y fácilmente extensibles a partir de componentes reutilizables.
- Disminución en el tiempo de desarrollo gracias a la reutilización del código.
- El desarrollo del software es más intuitivo porque las personas piensan naturalmente en términos de objetos más que en términos de algoritmos de software.

A continuación, se presenta una breve descripción de Java, el lenguaje de programación orientado a objetos que se usó en el desarrollo de este proyecto:

2.6.7 Java y JDK (Java Development Kit). Java es un lenguaje desarrollado por Sun Microsystems, en el año 2009 fue adquirida por la compañía Oracle.

El JDK es un software que provee herramientas de desarrollo para la creación de programas en java.

Para trabajar con Java se necesita un kit de desarrollo que proporciona:

- Un compilador: `javac`
- Un intérprete: `java`.
- Un generador de documentación: *javadoc*
- Un visor de applet para generar sus vistas previas, ya que un applet carece de método `main` y no se puede ejecutar con el programa `java`: `Appletviewer`.

2.7 SERVIDORES WEB

Es un tipo de software que se encuentra a la espera de una petición hecha por una aplicación cliente y da respuesta a dicha petición a través de una página web. Para cada transacción el servidor debe realizar dos acciones básicas: integrar todos los componentes de la página (texto, imágenes, vídeo, scripts, etc.) y enviarla rápidamente al usuario. A continuación, se describe el servidor Web que se ajusta a la tecnología escogida para el proyecto.

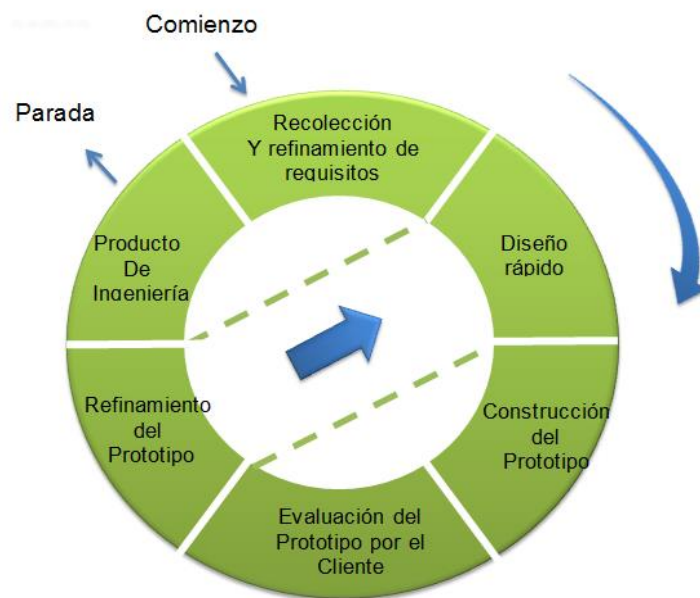
2.7.1 Servidor Jakarta Tomcat. Servidor de aplicaciones Java basado en los estándares definidos por Sun Microsystems. Tomcat es desarrollado como parte del proyecto de código abierto Jakarta de la fundación de software Apache y es uno de los servidores de aplicaciones Java más utilizados, en especial porque es liviano, cumple con todos los estándares, sencillo de instalar, tiene muy buena documentación y es gratuito, además por ser escrito en Java funciona en cualquier sistema operativo que disponga de la Máquina Virtual de Java (JVM).

3. MARCO METODOLÓGICO

3.1 PROTOTIPO EVOLUTIVO

Para realizar los nuevos servicios para los portales web comunidad académica de las diferentes escuelas y facultades de la Universidad Industrial de Santander se propone como metodología de desarrollo el prototipo evolutivo.

Figura 3 - Prototipo evolutivo



La elección de la metodología de prototipo evolutivo se debe a las siguientes razones:

- Es deseable tener un bosquejo de lo que se desee mejorar o crear para poder incorporar sugerencias de cambio por parte de los usuarios del portal de las escuelas en las etapas tempranas del desarrollo.
- Por otra parte, es necesario saber lo antes posible si hemos interpretado correctamente las especificaciones y las necesidades de las escuelas y de los profesores.

- En muchos casos los usuarios no tienen una idea definida de lo que desean, por lo tanto, debemos tomar decisiones y suponer qué es lo que el usuario quiere.
- Por este motivo, la emisión de los prototipos brinda la posibilidad de efectuar refinamientos de los requerimientos en forma sucesiva a fin de acercarse al producto deseado. Con el prototipo evolutivo se comienza diseñando y construyendo las partes más importantes de la aplicación en un prototipo que posteriormente se refina y ampliará hasta que el prototipo se termine. Este prototipo será el software que se entregará al final.
- La decisión se fundamenta en la ventaja de la realización de los cambios en etapas tempranas y la posibilidad de emisión de varios prototipos evaluables durante el desarrollo, obteniéndose de este modo, y de forma paralela, una metodología integral también para el proceso de evaluación del programa.
- Esta metodología propicia un intercambio de conocimientos y de autocrítica al sistema, lo que conlleva a que se produzcan muchas pruebas antes de liberar una nueva versión, así como mejoras rápidas a problemas que puedan surgir durante su uso.

Procedimiento que seguir para la metodología planteada:

- Hacer un análisis de los requerimientos para la construcción de los prototipos.
- Desglosar los objetivos globales con el fin de tener una idea más detallada del software a realizar, mediante reuniones entre los desarrolladores y los usuarios, en las cuales se identifican los requerimientos de los usuarios y se concluyen los aspectos que requieren una mayor definición.
- Presentar al usuario el diseño de un prototipo enfocado en los aspectos visuales del software, métodos de entrada y formatos de salida, para proceder a la construcción del mismo.
- Evaluación del prototipo por parte del usuario para filtrar los requisitos del software a desarrollar.

- Se produce un proceso interactivo en el que el prototipo es depurado para satisfacer necesidades del usuario, de igual forma el desarrollador obtiene una mejor comprensión de lo que hay que hacer para la entrega del producto final de ingeniería requerido por el usuario.

3.2 LENGUAJE DE MODELADO UNIFICADO

El Lenguaje de Modelado Unificado o Unified Modeling Language (UML), es el más utilizado en la actualidad. Es un lenguaje gráfico estándar para visualizar, especificar, construir y documentar un sistema para describir un modelo del sistema, incluyendo aspectos conceptuales tales como procesos de negocio, funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes reutilizables.

UML no es un método de desarrollo porque no indica los pasos a seguir para llegar al código, es decir, no especifica como pasar del análisis al diseño y de este al código. Al no ser un método de desarrollo resulta ser independiente del ciclo de desarrollo que se siga, puede encajar en un ciclo en cascada, evolutivo, espiral o en métodos ágiles de desarrollo.

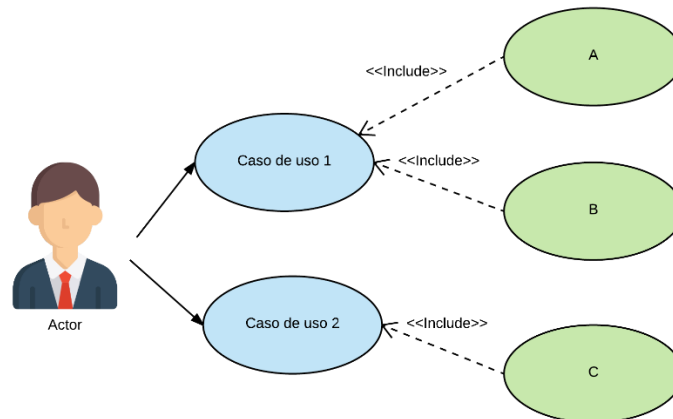
3.2.1 Diagramas de UML. Los diagramas UML utilizados en el desarrollo de este proyecto fueron diagramas de casos de uso y diagramas de secuencias. Las principales razones por las cuales se prefirió UML como el lenguaje de modelado son:

- UML tiene una notación gráfica muy expresiva que permite representar todas las fases de un proyecto informático, desde el análisis con casos de uso, el diseño con diagramas de clases, objetos, etc.

- UML facilita el entendimiento de la información, la función y el comportamiento de un sistema, haciendo fácil el análisis de los requerimientos, ya que sirve de apoyo en los procesos de análisis de un problema.
- UML permite a los creadores de sistemas realizar diseños que faciliten la comunicación a otras personas de manera convencional.
- UML permite generar un punto de comparación entre lo logrado y lo planificado.

3.2.2 Diagramas de casos de uso. Representación gráfica del entorno del sistema (actores) y su funcionalidad principal. Describe lo que hace el sistema desde el punto de vista de un observador externo, concentrándose en expresar lo que hace el sistema y no en dar respuesta de cómo lograr su comportamiento.

Figura 4 - Diagramas de Casos de Uso



3.2.2.1 Actores. Un actor en un caso de uso representa un rol, que alguien o algo puede desempeñar dentro un sistema y no un alguien o algo específico.

En este proyecto se destacan tres clases de actores:

- **Administradores:** Son usuarios que además de pertenecer a la categoría de usuarios tienen un perfil de administrador, con el cual tiene permisos extras a los que tiene un usuario comúnmente dentro del sitio; alguno de estos son los

auxiliares de administración del portal, los profesores, las secretarías de las escuelas, entre otras. Dentro de esta categoría se incluye también el súper administrador.

- **Súper Administrador:** Es el usuario que puede administrar, controlar y modificar los portales web de las escuelas, sus parámetros y sus usuarios.
- **Usuario Portal Web Comunidad Académica:** Es el tipo de usuario común de los portales web y a quien van dirigidas las páginas de servicio. Este usuario solo tiene control sobre sus servicios permitidos.

3.2.2.2 Inclusión (include). Es una forma de interacción, un caso de uso dado puede "incluir" otro. Una inclusión es utilizada para indicar que un caso de uso depende de otro, es decir, la funcionalidad de determinado caso de uso se requiere para realizar las tareas de otro. En la figura 4 el caso de uso "Caso de uso 1" depende de los casos de uso "A" y "B".

3.2.2.3 Extensión (extend). Es otra forma de interacción, una extensión representa una variación de un caso de uso a otro, es decir, una dependencia específica entre los casos de uso, a través de la cual un caso de uso puede extender a otro.

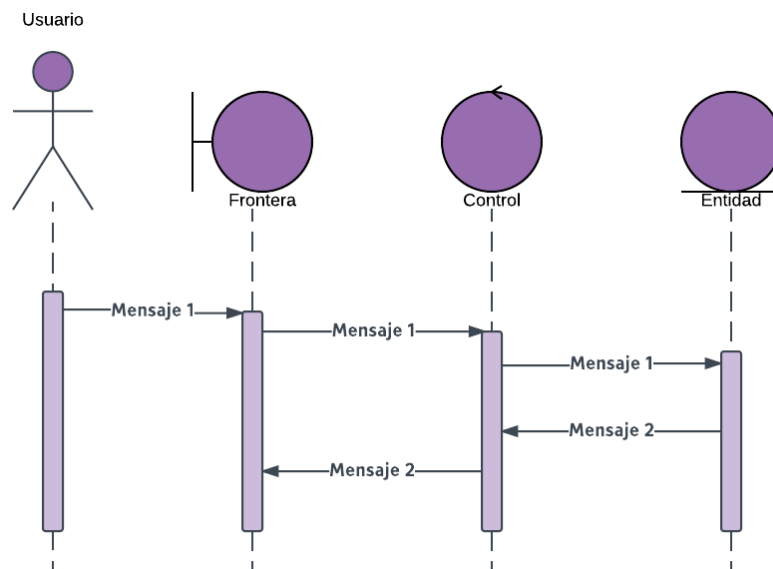
3.2.2.4 Diagramas de secuencias. Es aquel que muestra la forma en que los objetos interactúan entre sí al transcurrir el tiempo. Consta de objetos que se representan del modo usual: rectángulos con nombre (subrayado), mensajes representados por líneas continuas con una punta de flecha y el tiempo representado como una progresión vertical.

- **Objetos:** Se ubican en la parte superior del diagrama de izquierda a derecha y se acomodan de manera que simplifiquen al diagrama. La línea que está debajo de cada objeto será una línea discontinua conocida como la *línea de vida* de un

objeto. Con la línea de vida se encuentra un pequeño rectángulo conocido como *activación*, el cual representa la ejecución de una operación que realiza el objeto.

- **Mensaje:** Un mensaje que va de un objeto a otro pasa la línea de vida de un objeto a otro. Un objeto puede enviarse un mensaje a sí mismo. Un mensaje puede ser simple, sincrónico o asincrónico.
- **Tiempo:** El diagrama representa al tiempo en dirección vertical. Inicia en la parte superior y avanza hacia la parte inferior. Un mensaje que esté más cerca de la parte superior ocurrirá antes que uno que esté cerca de la parte inferior.
- **GUI:** (Siglas en Ingles) La interfaz gráfica de usuario; es la interfaz de interacción del usuario y en la que más interactividades se presentan con otros objetos.

Figura 5 - Diagrama de Secuencias



3.3 ESTÁNDARES DE PROGRAMACIÓN

3.3.1 Modelo de datos. Es un lenguaje utilizado para la descripción de una base de datos, por lo general permite describir estructuras de datos de la base de datos

(el tipo de datos que incluye la base y la forma en que se relacionan), las restricciones de integridad (las condiciones que los datos deben cumplir para reflejar correctamente la realidad deseada) y las operaciones de manipulación de los datos (agregar, borrar, modificar).

3.3.2 Nombres de las tablas. Los nombres de los campos, así como de las tablas de la base de datos, se escriben en minúsculas, exceptuando los prefijos TP, TR y TB que indican si la tabla es principal, relacional o básica, respectivamente, y exceptuando también la primera letra de cada palabra que conforme su nombre; si es un nombre compuesto por dos o más palabras, los nombres tendrán en mayúscula la primera letra de cada palabra que la forma.

Como se mencionó, se han definido tres categorías para las diferentes tablas que conforman la base de datos. Dada la categoría de la tabla, se agrega un prefijo a su nombre que permita conocer la categoría a la que pertenece. Las categorías son:

- **Tabla básica:** Aquella cuyos registros son necesarios para el correcto funcionamiento de la base de datos. Estas tablas no experimentan muchos cambios en los datos. El prefijo a anteponer a los nombres de estas tablas es “TB_”, es decir la tabla que almacena las categorías de clasificación de los usuarios del portal web es llamada “TB_Categorías”, por ejemplo.
- **Tabla de Relación:** Surge de la relación muchos a muchos de una o dos tablas cualquiera. Los nombres de las tablas de relación deben ser siempre descriptivos para cada relación. El prefijo a anteponer a los nombres de estas tablas es “TR_”, por ejemplo, la tabla “MiembrosConsejo” es conocida como “TR_MiembrosConsejo”.
- **Tabla Principal:** Aquella cuyo número de registros tiende a crecer en gran cantidad y que además no es posible clasificar como tabla básica o de relación. Un ejemplo de tabla principal es la tabla que almacena los usuarios del portal

EISIWeb. El prefijo a anteponer a los nombres de estas tablas es “TP_”, es decir la tabla “Usuarios”, es conocida como “TP_ Usuarios”.

3.3.3 Clases. Los nombres de las clases deben ser sustantivos en plural, la primera letra de cada palabra debe ser mayúscula. Estos deben ser simples, descriptivos como, por ejemplo: TipoPlan.java, Usuarios.java, NotasCartelera.java, etc.

3.3.4 Páginas JSP. Los nombres de las páginas JSP que componen los portales web comunidad académica son escritos de manera que la primera letra es una mayúscula seguido de letras minúsculas, en caso de que el nombre del JSP sea compuesto por dos o más palabras, entonces la primera de cada palabra debe ir en mayúscula, por ejemplo: ListarTipoPlan.jsp, PeticionesTipoPlan.jsp, ModificarTipoPlan.jsp, etc.

3.3.5 Organización de Directorios. Los directorios del sitio están organizados de tal manera que los archivos que se almacenen en ellos correspondan a lo que describe el nombre del directorio. Por ejemplo: Aula.

- El sitio cuenta con un directorio llamado “images”; en éste se encuentran almacenados todos los archivos .jpg, .gif, .png.
- Para el desarrollo de este proyecto se crearon los directorios “/archivos/usuario”, “/eisi/images/Aula/images”, “/eisi/images/Aula/archivos”, “/eisi/images/Aula/flash”, los cuales almacenan los archivos correspondientes al funcionamiento de estos servicios.

4. DESARROLLO DE LA HERRAMIENTA, ADMINISTRACIÓN Y MANTENIMIENTO

Para el desarrollo de este proyecto se siguió la metodología de prototipo evolutivo. Al iniciar el proyecto se elaboró un primer prototipo durante la fase de requerimientos, el cual fue mejorado con la inclusión de nuevos requerimientos surgidos en la fase de desarrollo, a medida que se generaba un prototipo, el mismo era sometido a pruebas de funcionamiento y se le realizaban los refinamientos pertinentes a partir del resultado de dichas pruebas.

4.1 PROTOTIPO ESPERADO

Al iniciar el proyecto aún no se contaba con una concepción clara de lo sería el producto final, sin embargo, en el transcurso del desarrollo y evolución de los prototipos, las pruebas y análisis del sistema se pudo comprobar que se estaba acercando a los requerimientos iniciales, esto con el fin de enfocar el desarrollo a la solución de las necesidades de los usuarios

El objetivo específico inicial y los requisitos que surgieron se dieron gracias a la realización de prototipos y la realimentación con el cliente. Para cada objetivo se listaron los requerimientos detallados de este, los cuales se cumplieron para el prototipo final.

4.1.1 Requerimientos de objetivo. El objetivo específico inicial y los requisitos que surgieron se dieron gracias a la realización de prototipos y la realimentación con el cliente. Para cada objetivo se listan los requerimientos detallados de este, los cuales se cumplieron para el prototipo final.

4.1.1.1 Creación de nuevas interfaces gráficas y actualización de las interfaces antiguas

Objetivo inicial:

- Cambiar el diseño para una mejor presentación de los resultados y usabilidad de los servicios desarrollados.

Requisitos finales del objetivo:

- Los usuarios se encontrarán con interfaces nuevas, agradables con mejor interacción usuario-servicio, actuales, bajo el estándar Elise*.
- Las nuevas interfaces dentro de los servicios de nuevos tipos de planes, notas, archivos y porcentaje de actividades son más intuitivas, sencillas y novedosas. Pensando en la facilidad de uso por parte del usuario y mejor adaptación a los cambios presentados.

4.1.1.2 Soportar la creación de nuevos tipos de planes, debido a que actualmente solo existen cuatro tipos, estos son: foros, trabajos, exposiciones y laboratorios. Debe poderse agregar cualquier otro tipo, y este debe comportarse correcto al activarse y evaluarse.

Objetivo inicial:

- Crear un nuevo servicio que permita crear nuevos tipos de planes, con una interfaz actual, diseño intuitivo que permita a los usuarios hacer uso del servicio fácilmente.

Requisitos finales del objetivo:

- Se encarga de crear nuevos tipos de planes, estos se pueden activar y desactivar, modificar ó eliminar.
- Una vez creado el nuevo tipo de plan, este permitirá agregarle actividades, y ser usado normalmente como cualquier otro tipo de plan.

* Estandarización de los proyectos COMA y RealTime. Se encuentra en:
<http://cormoran.uis.edu.co/eisi/Calumet/Estandar/>

- Al crear el nuevo tipo de plan se requerirá la información de este, como el título y la descripción.

4.1.1.3 Diseñar y desarrollar el servicio que permita modificar el contenido de los planes, como el cambio de estado entre Activo/Inactivo

Objetivo inicial:

Tener una interfaz que permita modificar, eliminar y cambiar estado de todos los tipos de planes.

Requisitos finales del objetivo:

- El usuario que tenga el poder de crear nuevos tipos de planes, al hacer uso del servicio podrá visualizar una tabla con los nuevos tipos de planes creados, su estado activo/inactivo, y opciones de modificar ó eliminar.
- Los tipos de planes antiguos, como son: Exámenes, talleres, foros, trabajos, laboratorios y exposiciones, se podrán modificar, activar o desactivar, pero no se podrán eliminar.

4.1.1.4 Realizar reingeniería en el Aula a las notas y archivos, que contemple mejora de la interfaz gráfica, así como de su funcionalidad

Objetivo inicial:

Cambios en la presentación y reingeniería de los servicios de notas y archivos.

Requisitos finales del objetivo:

- Se pueden crear notas por diferentes características: usuarios, subgrupos y grupos.
- Cualquier integrante del grupo, sea docente o estudiantes, podrán compartir notas con quienes deseen dentro del grupo.
- Se agregó la herramienta Filemanager en el servicio de notas dentro del aula, para permitir compartir imágenes.

- El usuario del aula podrá tener una vista general de notas en el aula organizada de la siguiente forma: una vista principal al momento de entrar al aula en donde aparecerán las notas para el usuario, al dar clic a la nota se abrirá una ventana en donde se visualizará la nota, una foto de la persona que la creó, la fecha, etc. Hay otra pestaña en donde se verán las notas creadas por el usuario, estas se verán en forma de tabla, se podrán modificar, publicar para un tiempo posterior, activar o desactivar o eliminar, además de visualizar los detalles de la nota.
- Se cambió la interfaz de compartir archivos dentro del aula, la nueva es más intuitiva, sencilla y novedosa. Se visualizarán dos pestañas, una en donde estarán los archivos compartidos con el usuario, aquí se podrán ver los archivos, descargar, al dar clic se verá la información detallada de este, como el título del archivo, descripción, quien lo compartió, etc. Hay otra pestaña en donde se verán los archivos compartidos por el usuario, su visualización es igual a la de la otra pestaña sólo que en esta se permite eliminar y modificar el archivo compartido.

4.1.1.5 Realizar reingeniería para que sea posible incluir el plan de quices al crear el aula

Objetivo inicial:

- Con la reingeniería se podrán agregar todos los tipos de planes que se creen y los quices. Además, estos pasarán a ser un plan por defecto.

Requisitos finales del objetivo:

- Actualmente al crear un aula solo se puede agregar los planes por defecto: Exámenes, Talleres, Trabajos, Laboratorios, Foros y Exposiciones.
- Se agregó como plan por defecto los quices, se permite asignar el tipo de calificación y el porcentaje de nota que representa en la definitiva.
- En el aula aparecerá el nuevo plan de quices en donde se permitirá agregar más, asignar fechas a realizar, calificar, etc.

4.1.1.6 Realizar reingeniería para que sea posible asignar porcentajes a las actividades

Objetivo inicial:

Cambiar el diseño de las actividades para permitir modificar el porcentaje de la nota.

Requisitos finales del objetivo:

- El docente podrá modificar los porcentajes de las actividades, ya no tendrán el mismo valor dentro del plan, ahora unos podrán valer más que otros dependiendo de la complejidad de la actividad.
- La nota definitiva de los estudiantes será asignada según los porcentajes de las actividades y se podrá visualizar en la tabla de notas que valor tenía cada actividad.
- Al seleccionar un tipo de plan en el aula, aparecen las actividades que pertenecen a este, además de los detalles de la actividad. Se creó una columna que contiene el porcentaje de la actividad, si no se han modificado los porcentajes entonces este será igual en todas las actividades dando una suma total de 100. Al modificar los porcentajes solo se permitirá guardar cuando la suma de los porcentajes de las actividades sea igual a 100.

4.2 DIAGRAMAS DE CASOS DE USO

4.2.1 Servicio creación de nuevos tipos de planes

Figura 6 - Diagrama de casos de uso: Crear servicio para nuevos tipos de planes.

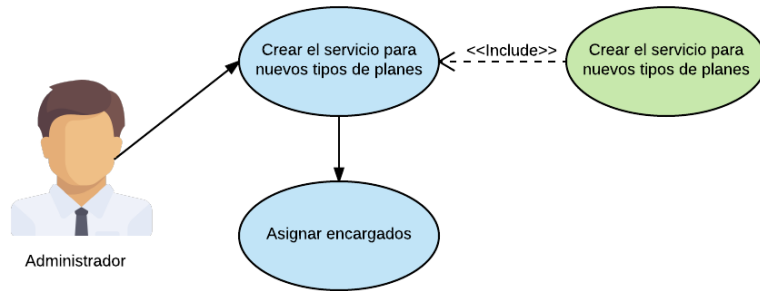


Figura 7- Diagrama de casos de uso: Creación de nuevos tipos de planes.

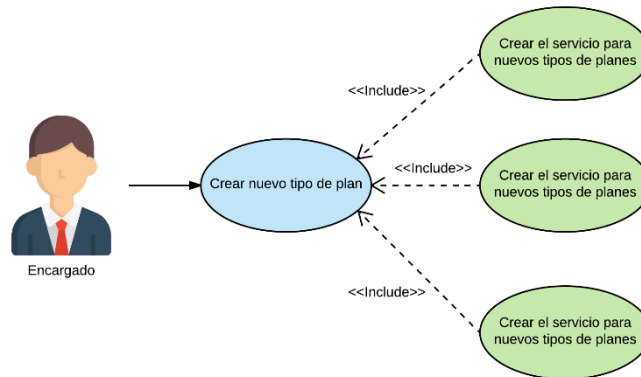


Figura 8 - Diagrama de casos de uso: Asignar actividades a nuevos tipos de planes.

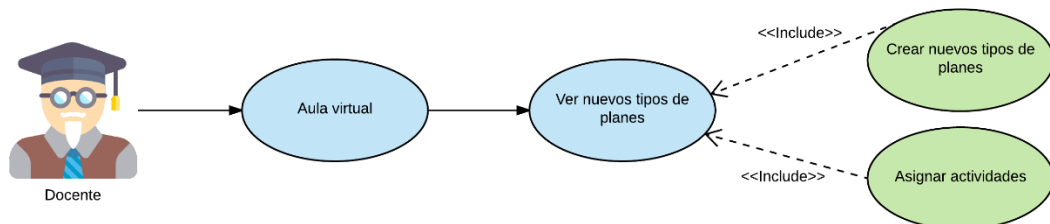


Figura 9 - Diagrama de casos de uso: Estudiante visualiza planes.

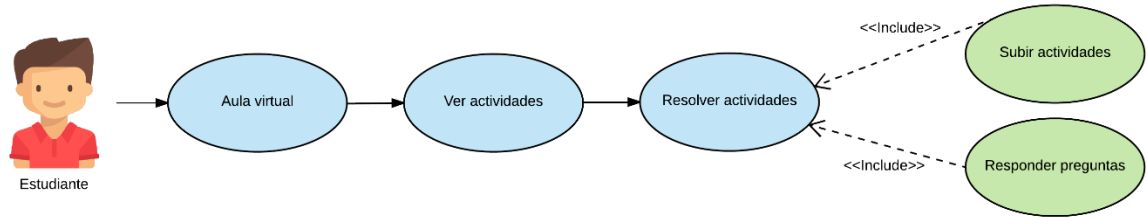


Figura 10 - Diagrama de casos de uso: Administración y creación de nuevos tipos de planes.

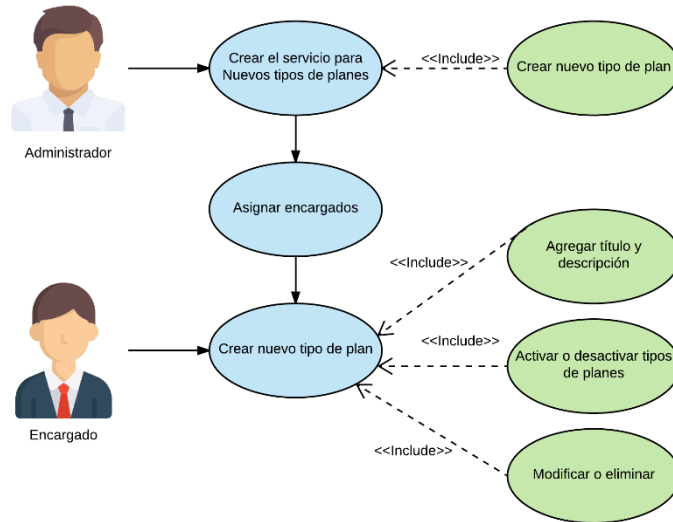
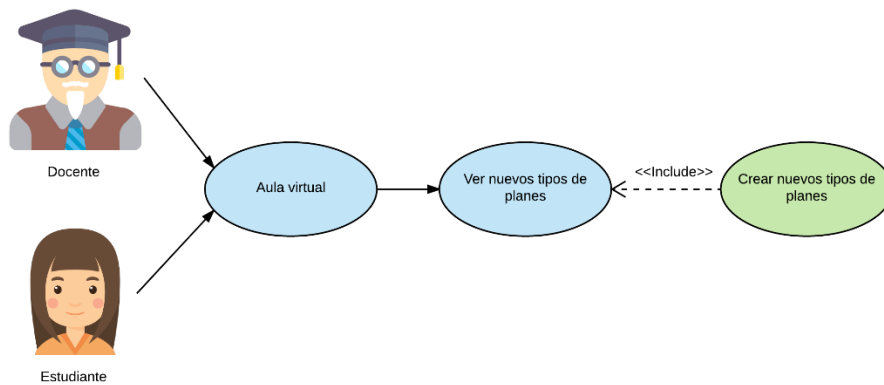


Figura 11 - Diagrama de casos de uso: Interacción docente – estudiante.



4.2.2 Servicio de notas en el Aula virtual

Figura 12 - Diagrama de casos de uso: Creación y visualización de las notas en cartelera.

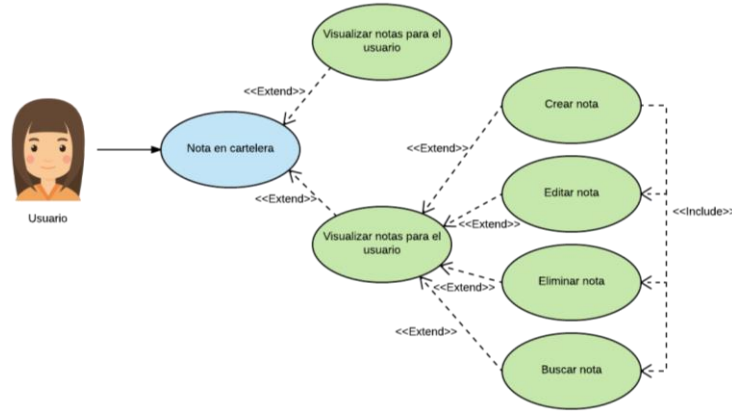
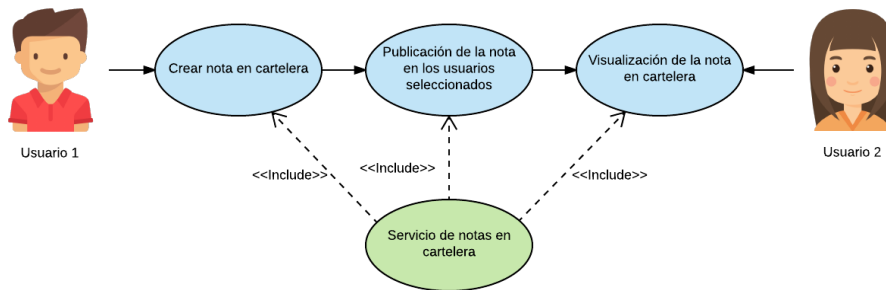


Figura 13 - Diagrama de casos de uso: Compartir notas entre usuarios.



4.2.3 Servicio de compartir archivos en el Aula virtual

Figura 14 - Diagrama de casos de uso: Creación y visualización de los archivos compartidos.

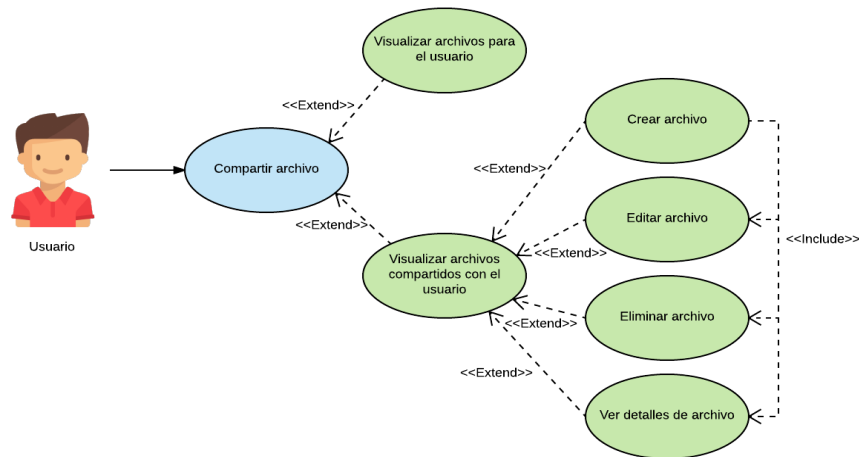
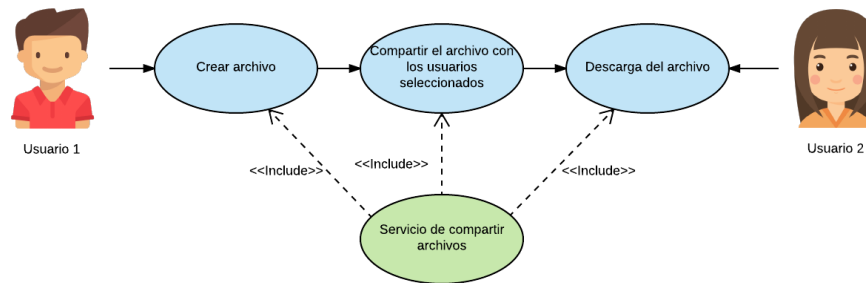
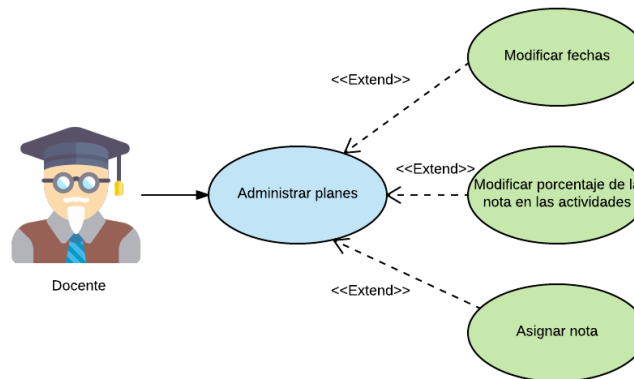


Figura 15 - Diagrama de casos de uso: Compartir archivos entre usuarios.



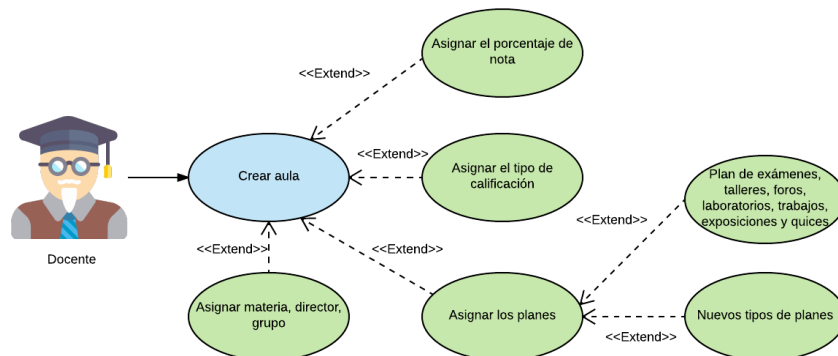
4.2.4 Servicio para asignar porcentajes a las actividades

Figura 16 - Diagrama de casos de uso: Modificar porcentajes de actividades.



4.2.5 Servicio para asignar porcentajes a las actividades

Figura 17 - Diagrama de casos de uso: Asignar planes al crear aula.



4.3 DOCUMENTACIÓN DE CASOS DE USO DEL SISTEMA

4.3.1 Servicio creación de nuevos tipos de planes

Tabla 1 - Casos de uso: Crear servicio para nuevos tipos de planes

TÍTULO	FUNCIONES PRIMARIAS
Crear el servicio para nuevos tipos de planes	El administrador del portal agrega el servicio para crear nuevos tipos de planes y lo añade a un menú en la sección del aula virtual.
Asignar encargados	Se asignan los perfiles de los usuarios que podrán ver este servicio.
Crear nuevo tipo de plan	Servicio que permite crear nuevos tipos de planes.

Tabla 2 - Casos de uso: Creación de nuevos tipos de planes.

TÍTULO	FUNCIONES PRIMARIAS
Crear nuevo tipo de plan	Servicio que permite crear nuevos tipos de planes, solo se visualiza por el usuario que tenga permisos.
Asignar título y descripción	Al crear el nuevo tipo de plan se debe asignar su título y descripción.
Activar o desactivar tipos de planes	Cambiar el estado de un tipo de plan (Activo o Inactivo)
Modificar o eliminar	Los tipos de planes creados se pueden modificar o eliminar, los tipos de planes

	por defecto no se permiten modificar o eliminar.
--	--

Tabla 3 - Casos de uso: Asignar actividades a nuevos tipos de planes

TÍTULO	FUNCIONES PRIMARIAS
Aula virtual	El docente accede al aula virtual.
Ver nuevos tipos de planes	Aparecen los nuevos planes junto a los antiguos.
Asignar actividades	El docente selecciona el nuevo plan para asignar las actividades. Se asignan actividades o se crean para este tipo de plan.
Crear nuevos tipos de planes	Servicio que permite crear y visualizar nuevos tipos de planes.

Tabla 4 - Casos de uso: Estudiante visualiza planes

TÍTULO	FUNCIONES PRIMARIAS
Aula virtual	El estudiante accede al aula virtual.
Ver actividades	En la página principal aparecen las actividades programadas para la semana de los diferentes planes.
Resolver actividades	El estudiante resuelve las actividades que le corresponden.
Responder preguntas	El estudiante puede responder preguntas si la actividad así lo requiere.
Subir archivos	Algunas actividades exigen subir archivos para ser calificados por el profesor.

Tabla 5 - Casos de uso: Administración y creación de nuevos tipos de planes

TÍTULO	FUNCIONES PRIMARIAS
Crear el servicio para nuevos tipos de planes	El administrador del portal agrega el servicio para crear nuevos tipos de planes y lo añade a un menú en la sección del aula virtual.
Asignar encargados	Se asignan los perfiles de los usuarios que podrán ver este servicio.
Crear nuevo tipo de plan	Servicio que permite crear nuevos tipos de planes.
Asignar título y descripción	Al crear el nuevo tipo de plan se debe asignar su título y descripción.
Activar o desactivar tipos de planes	Cambiar el estado de un tipo de plan (Activo o Inactivo).
Modificar o eliminar	Los tipos de planes creados se pueden modificar o eliminar, los tipos de planes por defecto no se permiten modificar o eliminar.
Crear nuevo tipo de plan	Servicio que permite crear nuevos tipos de planes, solo se visualiza por el usuario que tenga permisos.

Tabla 6 - Casos de uso: Interacción docente – estudiante

TÍTULO	FUNCIONES PRIMARIAS
Aula virtual	El docente o estudiante acceden al aula virtual.
Ver nuevos tipos de planes	En la pantalla principal aparecen los nuevos tipos de planes.

Crear nuevos tipos de planes	Servicio que permite crear nuevos tipos de planes.
------------------------------	--

4.3.2 Servicio de notas en el Aula virtual

Tabla 7 - Casos de uso: Creación y visualización de las notas en cartelera

TÍTULO	FUNCIONES PRIMARIAS
Nota en cartelera	Servicio que permite crear notas en cartelera.
Visualizar notas para el usuario	Pestaña que muestra las notas para el usuario.
Administrar notas creadas por el usuario	Pestaña en donde se permite administrar las notas del usuario.
Crear nota	El usuario crea una nota para un/unos usuarios de la materia.
Editar nota	Se permite modificar las notas creadas.
Eliminar nota	Se permite eliminar las notas que no se necesitan.
Buscar nota	Filtro de la tabla para buscar una nota específica.

Tabla 8 - Casos de uso: Compartir notas entre usuarios

TÍTULO	FUNCIONES PRIMARIAS
Crear nota en cartelera	Un usuario crea una nota en cartelera dirigida a otro usuario.

Publicación de la nota en los usuarios seleccionados	El servicio de crear notas en cartelera asigna la nota a la persona indicada.
Visualización de la nota en cartelera	Cuando el usuario al que va dirigida la nota entra al aula, este verá la nota dirigida hacia él en la pantalla principal.
Servicio de notas en cartelera	Servicio que permite publicar notas en cartelera.

4.3.3 Servicio de compartir archivos en el Aula virtual

Tabla 9 - Casos de uso: Creación y visualización de los archivos compartidos

TÍTULO	FUNCIONES PRIMARIAS
Compartir archivo	Servicio que permite crear archivos en el aula.
Visualizar archivos para el usuario	Pestaña que muestra los archivos para el usuario.
Administrar archivos compartidos por el usuario	Pestaña en donde se permite administrar los archivos del usuario.
Crear archivo	El usuario crea un archivo para un/unos usuarios de la materia.
Editar archivo	Se permite modificar los archivos creadas.
Eliminar archivo	Se permite eliminar los archivos que no se necesitan.
Ver detalles de archivo	Al seleccionar un archivo este muestra los detalles como título, descripción, número de descargas, entre otras.

Tabla 10 - Casos de uso: Compartir archivos entre usuarios

TÍTULO	FUNCIONES PRIMARIAS
Crear archivo	Un usuario crea un archivo en el aula dirigida a otro usuario.
Compartir el archivo con los usuarios seleccionados	El servicio de crear archivos en el aula asigna el archivo a la persona indicada.
Descarga del archivo	Cuando el usuario al que va dirigida el archivo entra al aula, este verá el archivo dirigida hacia él en la pantalla principal.
Servicio de compartir archivos	Servicio que permite publicar archivos en el aula.

4.3.4 Servicio para asignar porcentajes a las actividades

Tabla 11 - Casos de uso: Modificar porcentajes de actividades

TÍTULO	FUNCIONES PRIMARIAS
Administrar planes	El docente visualiza el plan al que desea modificar los porcentajes de sus actividades.
Asignar nota	El docente dentro del poder de administrar sus planes puede asignar las notas de las actividades según los estudiantes que hayan respondido.
Modificar porcentaje de la nota en las actividades	Se permite modificar el porcentaje de la nota de las actividades dentro de un plan.

Modificar fechas	El docente puede modificar las fechas de publicación de las actividades.
------------------	--

4.3.5 Realizar reingeniería para que sea posible incluir el plan de quices al crear el aula

Tabla 12 - Casos de uso: Asignar planes al crear aula

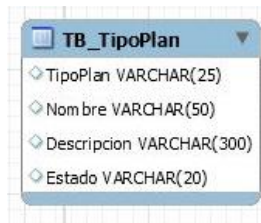
TÍTULO	FUNCIONES PRIMARIAS
Crear aula	El docente accede al servicio de crear un aula.
Asignar materia	El docente asigna la materia a la que pertenecerá el aula.
Asignar el porcentaje de nota	El docente asigna los porcentajes de la nota de los planes.
Asignar el tipo de calificación	El docente escoge el tipo de calificación de los planes, si es por actividad o por plan.
Asignar los planes	El docente asigna los planes del aula a los tipos de planes existentes.
Plan de exámenes, talleres, foros, laboratorios, trabajos, exposiciones y quices.	Tipos de planes por defecto al crear el aula.
Nuevos tipos de planes.	Si se ha creado un nuevo tipo de plan este aparecerá al momento de crear el aula.

4.4 DISEÑO Y ANÁLISIS

Diagramas Entidad/Relación de los servicios desarrollados

4.4.1 Servicio creación de nuevos tipos de planes

Figura 18 - Diagrama E/R: Creación de nuevos tipos de planes



4.4.2 Servicio de notas en el Aula virtual

Figura 19 - Diagrama E/R: Notas en el Aula virtual



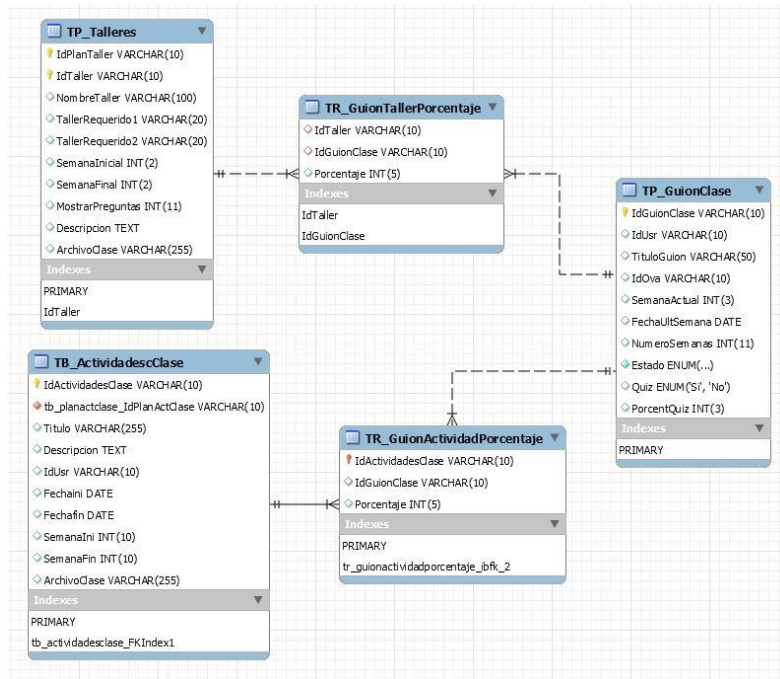
4.4.3 Servicio de compartir archivos en el Aula virtual

Figura 20 - Diagrama E/R: Compartir archivos en el Aula virtual



4.4.4 Servicio para asignar porcentajes a las actividades

Figura 21 - Diagrama E/R: Asignar porcentajes a las actividades



4.5 DESCRIPCIÓN DE LAS ENTIDADES

Descripción de las entidades creadas y utilizadas en las bases de datos Diamante ubicada en el servidor de cada portal y Poseidón ubicada en el servidor del portal web de la Vicerrectoría Académica.

Tabla 13 - Descripción de las entidades

ENTIDAD	DESCRIPCION
TB_TipoPlan	Contiene la información de los tipos de planes existentes en el portal de una escuela.
TP_CarteleraAulaVirtual	Contiene la información de las notas compartidas actualmente entre los usuarios del aula virtual del portal de una escuela.
TP_AulaArchivos	Contiene la información de los archivos compartidos actualmente entre los usuarios del aula virtual del portal de una escuela.
TP_Talleres	Contiene la información de los talleres y exámenes existentes en las aulas virtuales del portal.
TR_GuionTallerPorcentaje	Contiene los porcentajes de la nota asignados a los talleres y exámenes del aula virtual.
TP_GuionClase	Contiene los títulos asignados a las aulas virtuales existentes en el portal de una escuela.
TR_GuionActividadPorcentaje	Contiene los porcentajes de la nota asignados a las actividades del aula virtual.

TB_ActividadesClase	Contiene la información de las actividades existentes en las aulas virtuales del portal.
---------------------	--

4.6 MODELO DE PROCESOS DEL SISTEMA

Para una mejor interpretación de los modelos de los procesos del sistema se realizaron los diagramas de secuencia necesarios para cada caso de uso en los que se explica con detalle los pasos para el funcionamiento de cada uno de los servicios (Ver anexo A).

4.7 IMPLEMENTACIÓN, IMPLANTACIÓN Y PRUEBAS GENERALES

Para la implementación de los servicios se utilizaron las siguientes herramientas:

- Lenguaje de marcado para la elaboración de páginas web, HTML.
- Lenguaje de programación orientado a la web, JSP.
- Lenguaje de programación interpretado orientado a objetos, JavaScript.
- Hojas de estilo en cascada, CSS.
- Lenguaje Java.
- NetBeans, IDE para desarrollar las clases de Java y los archivos JSP.
- Servidor Jakarta Tomcat.
- Manejador de Base de datos, MySQL 5.0.

Haciendo uso de las anteriores herramientas, y con la asesoría y seguimiento del director de proyecto se diseñaron los nuevos servicios y se estructuraron los datos para el primer prototipo. También se tomaron en cuenta sugerencias recibidas por miembros del Grupo Calumet, para realizar un posterior refinamiento a las interfaces.

Para el desarrollo se trabajó con la base de datos "Diamante" ubicada en todos los servidores de los portales de las escuelas, y con la base de datos centralizada

"Poseidón" ubicada en el servidor del portal web de la Vicerrectoría Académica. En las cuales se crearon las tablas necesarias para el funcionamiento de los servicios.

Se llevaron a cabo pruebas para cada subsistema propuesto, verificando que el resultado correspondiera con lo esperado, de esta forma, se evidenció el correcto funcionamiento en la captura de datos, selección de ítems y almacenamiento de información.

La implantación de los servicios se realizó en primer lugar en el portal web de pruebas del Grupo Calumet, donde se realizan las pruebas a todos los componentes. Finalmente se implementan en todos los portales web a los cuales el Grupo Calumet presta soporte.

4.8 MANTENIMIENTO Y ADMINISTRACIÓN

4.8.1 Actividades de Mantenimiento. Dentro de las funciones que se realizan en la administración y mantenimiento de los servicios de los portales web se encuentra la tarea de corregir errores, las cuales se denominan incidencias, que se corrigen durante la primera fase como integrantes del Grupo Calumet.

4.8.2 Actividades de Soporte a Usuarios. Las escuelas cuentan en sus portales con un servicio de consultas y sugerencias, a través del cual se responden preguntas y se resuelven problemas de los usuarios. Diariamente se reciben consultas de estudiantes y profesores, que necesitan asesoría en el funcionamiento de servicios o soporte en el manejo de su usuario y contraseña. También se resuelven problemas de manera presencial, donde el usuario acude directamente a las oficinas del Grupo Calumet y es atendido directamente por alguno de sus

miembros, que le prestan la asesoría necesaria. Algunas de las labores realizadas comúnmente son:

- Restablecimiento de contraseña a usuarios.
- Dar respuesta y solución a las inquietudes y problemas que tengan los usuarios de los portales respecto al uso de los servicios.
- Facilitar orientación a los estudiantes nuevos de las escuelas a las cuales presta sus servicios el grupo Calumet, en cuanto al registro en el portal y el uso de sus principales servicios.

4.8.3 Actividades de Administración. Dentro de las actividades realizadas por los miembros del Grupo Calumet, se encuentra la tarea de administrar los servidores de los portales de las escuelas, cada semestre se asignan nuevos administradores, los cuales cuentan con un usuario dentro de uno de los portales a los cuales presta soporte el grupo. Como administrador las tareas que se realizan comúnmente son:

- Realizar periódicamente copias de las bases de datos Diamante y División.
- Dar aval a las solicitudes de publicación de los usuarios en la cartelera para que puedan ser accedidas por la comunidad.
- Atender las sugerencias hechas por los usuarios del sistema a través del servicio de consultas y sugerencias.
- Actualizar periódicamente las bases de datos con respecto a la información que ofrece la División de Servicios de Información de la Universidad Industrial de Santander, para que el portal cuente con información actualizada.

5. PRUEBAS DEL SISTEMA

Para garantizar el correcto desarrollo de los nuevos servicios creados y a los que se le hizo reingeniería, se realizaron las siguientes pruebas:

5.1 PRUEBAS DE VERIFICACIÓN

Esta prueba es una de las más utilizadas en desarrollo de software mediante esta se aplican diferentes técnicas para detectar errores en el sistema antes de ser utilizado.

Se efectúa ejecutando paso a paso el proceso del servicio de manera que se explora cada funcionalidad que tiene el módulo desarrollado, realizando verificaciones de validación, los campos que son obligatorios no pueden quedar vacíos, por ejemplo.

Las siguientes tablas describen las pruebas de cada caso de uso de los servicios que fueron desarrollados.

5.1.1 Pruebas por componente. Esta prueba se realizó para cada caso de uso de cada servicio desarrollado y descrito anteriormente:

5.1.1.1 Servicio creación de nuevos tipos de planes

Tabla 14 - Pruebas Realizadas: Creación de servicio nuevos tipos de planes

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Crear el servicio para nuevos tipos de planes	El administrador del portal agrega el servicio para crear nuevos tipos de planes y lo añade a un menú en la sección del aula virtual.	✓
Asignar encargados	Se asignan los perfiles de los usuarios que podrán ver este servicio.	✓

Tabla 15 - Pruebas Realizadas: Creación de nuevos tipos de planes.

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Asignar título y descripción	Se asigna un título y descripción a un tipo de plan que se crea.	✓
Activar o desactivar tipos de planes	Cambiar el estado del tipo de plan creado a Activo y luego a Inactivo. Se confirma que esto suceda en todos los lugares del aula.	✓
Modificar	Se modifica el título o la descripción del plan. Se confirma que esto suceda	✓

	en todos los lugares del aula.	
Eliminar	Se elimina el plan. Se confirma que esto suceda en todos los lugares del aula.	✓

Tabla 16 - Pruebas Realizadas: Asignar actividades a nuevos tipos de planes

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ver nuevos tipos de planes	Ver los nuevos planes junto a los antiguos.	✓
Asignar actividades	El docente selecciona el nuevo plan para asignar las actividades. Se asignan actividades o se crean para este tipo de plan.	✓

Tabla 17 - Pruebas Realizadas: Estudiante visualiza planes

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Ver actividades	En la página principal aparecen las actividades programadas para la semana de los diferentes planes, entre esos las de los nuevos tipos de planes.	✓

5.1.1.2 Servicio de notas en el Aula virtual

Tabla 18 - Pruebas Realizadas: Creación y visualización de las notas en cartelera

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Crear nota	Se crea una nota para un/unos usuarios de la materia.	✓
Editar nota	Se permite modificar las notas creadas.	✓
Eliminar nota	Se permite eliminar las notas.	✓
Buscar nota	Se hace filtro de la tabla para buscar una nota específica.	✓

Tabla 19 - Pruebas Realizadas: Compartir notas entre usuarios

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Crear nota en cartelera	Un usuario crea una nota en cartelera dirigida a otro usuario.	✓
Visualización de la nota en cartelera	El usuario al que va dirigida la nota la encuentra en la pantalla principal del aula.	✓

5.1.1.3 Servicio de compartir archivos en el Aula virtual

Tabla 20 - Pruebas Realizadas: Creación y visualización de los archivos compartidos

CASO DE USO	PRUEBA REALIZADA	RESULTADO
--------------------	-------------------------	------------------

Crear archivo	Se crea un archivo para un/unos usuarios de la materia.	✓
Descarga del archivo	El usuario al que va dirigido el archivo entra al aula, este ve el archivo dirigida hacia él en la pantalla principal.	✓
Editar archivo	Se permite modificar los archivos creados.	✓
Eliminar archivo	Se permite eliminar los archivos que no se necesitan.	✓
Ver detalles de archivo	Al seleccionar un archivo este muestra los detalles como título, descripción, número de descargas, entre otras.	✓

5.1.1.4 Servicio para asignar porcentajes a las actividades

Tabla 21 - Pruebas Realizadas: Modificar porcentajes de actividades

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Modificar porcentaje de la nota en las actividades	Se permite modificar el porcentaje de la nota de las actividades dentro de un plan.	✓

Volver a poner el porcentaje por defecto	Se permite volver a poner todos los porcentajes con el mismo valor.	✓
--	---	---

5.1.1.5 Asignar el plan de quices al crear el aula

Tabla 22 - Pruebas Realizadas: Agregar plan de quices al crear aula

CASO DE USO	PRUEBA REALIZADA	RESULTADO
Asignar materia	Asignar la materia a la que pertenecerá el aula.	✓
Asignar el porcentaje de nota	Asignar los porcentajes de los planes.	✓
Asignar el tipo de calificación	Escoger el tipo de calificación de los planes, si es por actividad o por plan.	✓
Asignar los planes	Asignar los planes del aula a los tipos de planes existentes.	✓
Plan de exámenes, talleres, foros, laboratorios, trabajos, exposiciones y quices. Nuevos tipos de planes.	Verificar que aparezcan los tipos de planes creados y los planes por defecto.	✓
Guardar	Guardar cambios y confirmar en el aula creada.	✓

6. CONCLUSIONES

1. La creación de nuevas interfaces gráficas y actualización de las interfaces antiguas permite una mejor interacción entre servicio y usuario, ayudando a un mejor entendimiento del servicio y mejorando la presentación del portal.
2. El desarrollo del servicio que permite la creación de nuevos tipos de planes facilita a los docentes del aula una personalización en los planes que manejan y elimina el límite de sólo seis tipos de planes que existían. Además, se permite modificar el estado entre Activo/Inactivo, en el caso donde no se desee eliminar un plan.
3. La reingeniería en el Aula a las notas y archivos mejoró su funcionalidad, su presentación y utiliza un gestor de archivos actual que permite agregar imágenes y archivos en el caso de las notas.
4. La reingeniería que permite incluir el plan de quices al crear el aula nos facilita implementar dentro del aula el uso de los quices, y que estos tengan un valor en la nota definitiva junto con los demás planes.
5. La reingeniería que permite asignar porcentajes a las actividades permite personalizar los valores de la nota de una actividad dentro de un plan, antes la nota de las actividades de un plan era el mismo, ahora el usuario puede variar el porcentaje de esta nota según lo necesite basándose en la complejidad de la actividad, tiempo de desarrollo, importancia, etc. Aumentando la comodidad del docente para sacar su nota definitiva.

7. RECOMENDACIONES

- Para mantener una buena comunicación con los usuarios de los portales, es fundamental para dar pronta y precisa respuesta a las sugerencias de los mismos dando un soporte oportuno a sus inquietudes.
- Capacitar a los usuarios sobre los cambios realizados en el servicio de crear nuevos tipos de planes para que así puedan hacer uso de manera óptima de este servicio.
- Capacitar a los usuarios del aula sobre los nuevos servicios implementados, crear notas en cartelera y compartir archivos, para que puedan hacer buen uso del mismo.
- Capacitar a los usuarios sobre las mejoras realizadas en el servicio de cambiar porcentajes de actividades para que puedan hacer buen uso del mismo y ahorren tiempo al ejecutar tareas relacionadas con este.

BIBLIOGRAFÍA

JAVASCRIPT-YA. Tutoriales del lenguaje de programación JavaScript. [En línea]. [Citado 26 enero, 2017]. <URL: <http://www.tutorialesprogramacionya.com/javascriptya/index.php?inicio=0>>.

JOHNSON, James. Bases de datos: Modelos lenguajes y diseño. 1ª ed. Oxford, 2000. Presenta temas de teoría de bases de datos: modelos y métodos de acceso, administración, diseño de aplicaciones.

MANUALES. Manuales básicos de JSP. [En línea]. [Citado 18 enero, 2017]. Disponible en Internet: <URL: <http://www.desarrolloweb.com/manuales/73/> >.

PROGRAMACIÓN. En este sitio se encuentran un completo curso para aprender java. [En línea]. [Citado 23 enero, 2017]. <URL: http://www.aprenderaprogramar.com/index.php?option=com_content&view=category&id=68&Itemid=188 >.

PROGRAMACIÓN FÁCIL. Manual de programación para Java Jsp (Java Server Pages). [En línea]. [Citado 16 Septiembre, 2016]. <URL:http://www.programacionfacil.com/programacion:manual_java_jsp>.

STALLINGS, William. Sistemas Operativos: Aspectos Internos y Principios de Diseño. Madrid: Pearson Prentice Hall, 2005. Este libro se ocupa de los conceptos completos de las características de los sistemas operativos.

STUMPF, Robert, TEAGUE, Lavette. Object-Oriented Systems Analysis and Design with UML. Prentice Hall. 2004. Este libro introduce los conceptos y métodos del análisis y diseño de sistemas orientados a objetos.

SQL. En este sitio se encuentra una buena documentación con ejemplos de sql. [En línea]. [Citado 20 enero, 2017]. <URL: <http://www.1keydata.com/es/sql/> >

WEITZENFELD, Alfredo. Ingeniería de Software Orientada a Objetos con UML, JAVA e Internet. Thomson International, 2003. En este libro se encuentra información sobre desarrollo de software orientado a objetos.

W3SCHOOLS. Tutoriales de cómo programar en html de gran ayuda en detalles básicos y sencillos. [En línea]. [Citado 20 enero, 2017]. <URL: <http://www.w3schools.com/html/>>.

WIKISPACES. Información acerca de la Arquitectura Cliente Servidor. [En línea]. [Citado 13 enero, 2017]. Disponible en Internet: <URL:<http://g701giadar.wikispaces.com/Arquitctura+Cliente+Servidor>>.

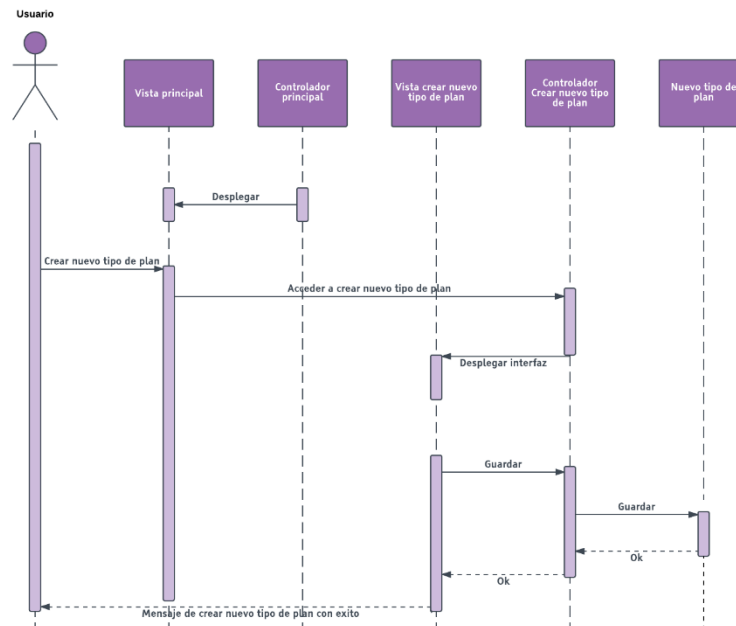
YOUBLISHER. Tutoriales básicos sobre UML. [En línea]. [Citado 18 enero, 2017]. <URL: <http://www.tutorialspoint.com/uml/> >

ANEXOS

Anexo A. Modelo de Procesos del Sistema

- Servicio creación de nuevos tipos de planes

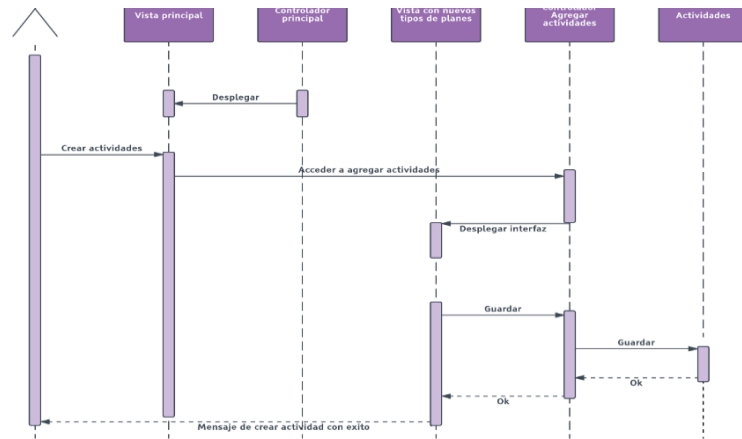
Figura 22. Diagrama de secuencia: Creación de nuevos tipos de planes



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crear un nuevo tipo de plan en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear nuevo tipo de plan al Controlador.
4. El controlador Crear nuevo tipo de plan recibe la petición y despliega la interfaz en la vista crear.
5. El Usuario quiere guardar los datos del tipo de plan.
6. La Vista crear nuevo tipo de plan envía la petición al Controlador crear y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador Crear nuevo tipo de plan y la Vista crear nuevo tipo de plan muestra un mensaje.

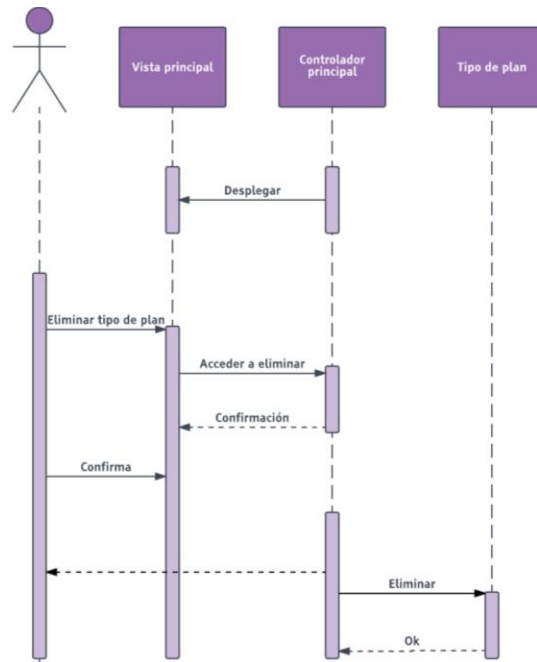
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.

Figura 23. Diagrama de secuencia: Asignar actividades a nuevos tipos de planes



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El usuario solicita crear una actividad para un nuevo tipo de plan.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de agregar actividades al Controlador.
4. El controlador Agregar actividades recibe la petición y despliega la interfaz en la vista con nuevos tipos de planes.
5. El Usuario quiere guardar los datos de las actividades.
6. La Vista con nuevos tipos de planes envía la petición al Controlador crear y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador Agregar actividades y la Vista con nuevos tipos de planes muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.

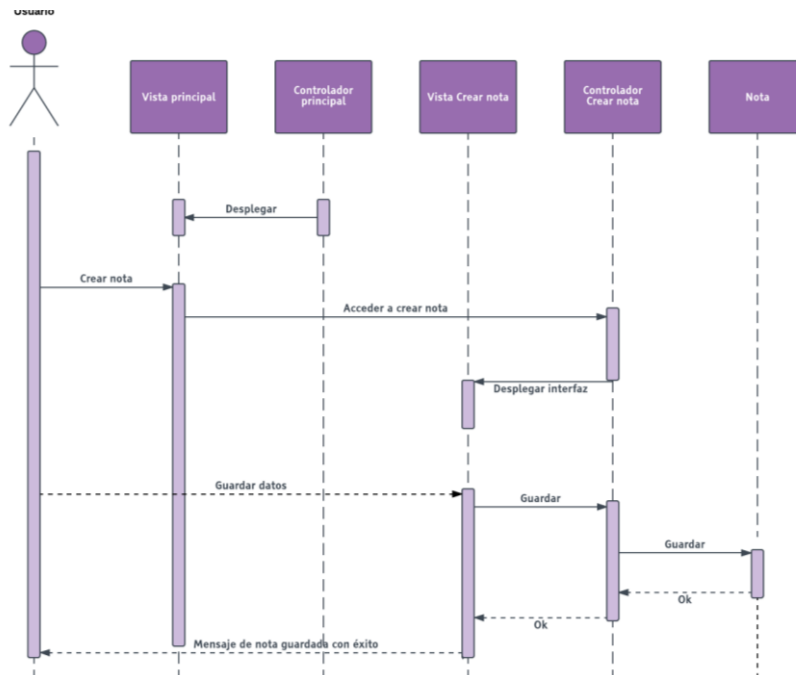
Figura 24. Diagrama de secuencia: Eliminación de tipo de plan



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El usuario solicita eliminar un plan.
3. El Controlador Principal envía un mensaje de confirmación por seguridad.
4. El Usuario confirma que quiere eliminar el plan.
5. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
6. El Controlador Principal solicita a la base de datos eliminar el plan.
7. La base de datos confirma que se eliminó el plan.
8. El controlador confirma que se eliminó y actualiza Vista Principal.

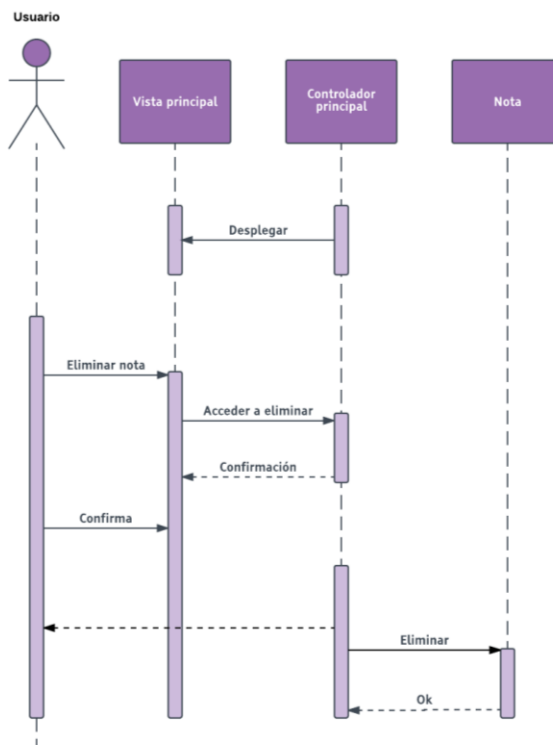
- **Servicio de notas en el Aula virtual**

Figura 25. Diagrama de secuencia: Creación de las notas en cartelera



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crear una nota en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear nota al Controlador.
4. El controlador Crear nota recibe la petición y despliega la interfaz en la vista crear.
5. El Usuario quiere guardar los datos de la nota.
6. La Vista crear nota envía la petición al Controlador crear nota y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador Crear nota y la Vista crear nota muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.

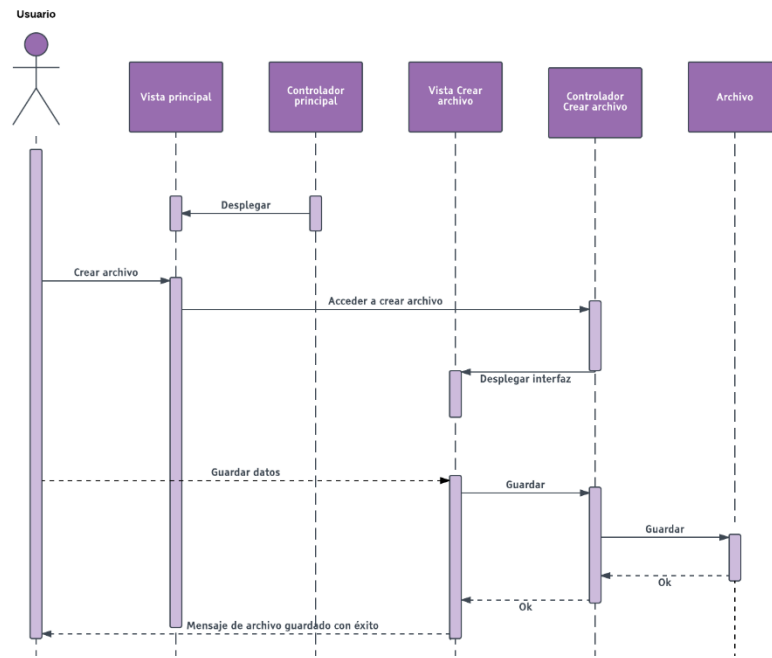
Figura 26. Diagrama de secuencia: Eliminar notas



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El usuario solicita eliminar una nota.
3. El Controlador Principal envía un mensaje de confirmación por seguridad.
4. El Usuario confirma que quiere eliminar la nota.
5. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
6. El Controlador Principal solicita a la base de datos eliminar la nota.
7. La base de datos confirma que se eliminó la nota.
8. El controlador confirma que se eliminó y actualiza Vista Principal.

- **Servicio de compartir archivos en el Aula virtual**

Figura 27. Diagrama de secuencia: Creación de archivos compartidos

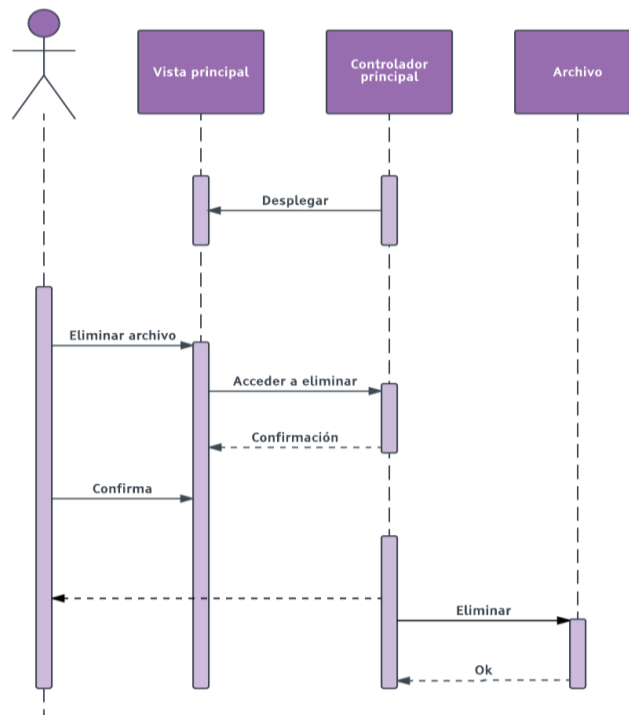


1. El

Controlador Principal despliega la interfaz en la Vista Principal.

2. El Usuario solicita crear un archivo en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de crear archivo al Controlador.
4. El controlador Crear archivo recibe la petición y despliega la interfaz en la vista crear.
5. El Usuario quiere guardar los datos del archivo.
6. La Vista crear archivo envía la petición al Controlador crear y el controlador hace la petición de guardar a la base de datos.
7. La base de datos guarda la información y da una respuesta de OK al Controlador Crear archivo y la Vista crear archivo muestra un mensaje.
8. El Usuario ve el mensaje de guardado y oprime aceptar para terminar.

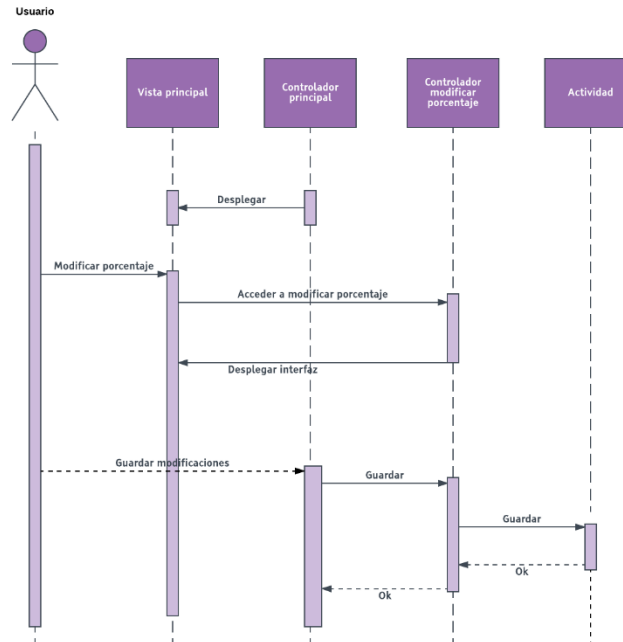
Figura 28. Diagrama de secuencia: Eliminar archivos compartidos



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El usuario solicita eliminar un archivo.
3. El Controlador Principal envía un mensaje de confirmación por seguridad.
4. El Usuario confirma que quiere eliminar el archivo.
5. La Vista Principal recibe la confirmación y la pasa al Controlador Principal.
6. El Controlador Principal solicita a la base de datos eliminar el archivo.
7. La base de datos confirma que se eliminó el archivo.
8. El controlador confirma que se eliminó y actualiza Vista Principal.

- **Servicio para asignar porcentajes a las actividades**

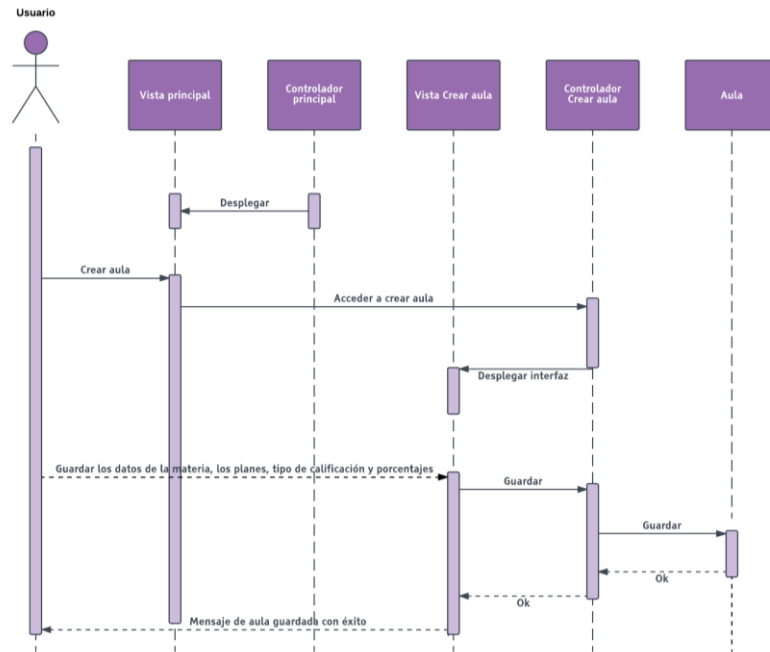
Figura 29. Diagrama de secuencia: Asignar porcentajes de actividades



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita modificar un porcentaje en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de modificar porcentaje al Controlador.
4. El controlador modificar porcentaje recibe la petición y despliega la interfaz en la vista principal.
5. El Usuario desea guardar los porcentajes modificados de las actividades.
6. El controlador principal revisa que la suma de los porcentajes sea igual a 100 y permite guardar los cambios.
7. El controlador modificar porcentaje hace una petición para guardar los cambios en la base de datos.
8. La base de datos actualiza la información y da una respuesta de OK al Controlador modificar porcentaje.
9. La página se actualiza con los porcentajes modificados.

- **Agregar plan de quices al crear aula**

Figura 30. Diagrama de secuencia: Crear aula



1. El Controlador Principal despliega la interfaz en la Vista Principal.
2. El Usuario solicita crea un aula en la Vista Principal.
3. La Vista Principal le envía la solicitud de acceder a la interfaz de Crear aula al Controlador.
4. El controlador Crear aula recibe la petición y despliega la interfaz en la vista principal.
5. El Usuario desea guardar los datos del aula, planes, tipo de calificación y porcentajes.
6. El controlador principal revisa que la suma de los porcentajes sea igual a 100 y permite guardar los cambios.
7. El controlador Crear aula hace una petición para guardar los cambios en la base de datos.
8. La base de datos actualiza la información y da una respuesta de OK al Controlador Crear aula.